



Interactive computer graphics in non-linear optimization.

GHARIB, Sheida.

Available from the Sheffield Hallam University Research Archive (SHURA) at:

<http://shura.shu.ac.uk/19683/>

A Sheffield Hallam University thesis

This thesis is protected by copyright which belongs to the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Please visit <http://shura.shu.ac.uk/19683/> and <http://shura.shu.ac.uk/information.html> for further details about copyright and re-use permissions.

101 480 038 2



Sheffield City Polytechnic
Eric Mensforth Library

REFERENCE ONLY

This book must not be taken from the Library

PL/26

R5193

ProQuest Number: 10696982

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10696982

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

INTERACTIVE COMPUTER GRAPHICS IN
NON-LINEAR OPTIMIZATION

by

SHEIDA GHARIB

Being a thesis submitted to the Council for National Academic
Awards in partial fulfilment of the requirements for the
Council's degree of

Master of Philosophy

Sponsoring establishment:

Sheffield City Polytechnic

Collaborating establishment:

University of Bradford

October, 1979

ACKNOWLEDGEMENT

The author wishes to express her thanks to her supervisors. Thanks are especially due to Dr I Cooper who has given valuable and constructive advice, and encouragement during the course of the study. The help received from Dr G F Raggett and Dr J Stephenson (University of Bradford) and Dr B G Hutley (Chemistry) is also acknowledged.

The author would also like to express her gratitude to the staff of Sheffield City Polytechnic Computer Services department.

Thanks are due to the Optimization Centre of Hatfield Polytechnic who contributed some ideas and material in the early stages of the project.

Finally the author is indebted to Mrs C J Barker for her skilful typing of this thesis.

ABSTRACT

S Gharib - "Interactive Computer Graphics in Non-linear Optimization"

The thesis surveys the current state of knowledge in the field of both interactive computer graphics and non-linear optimization.

The potential contribution of interactive computer graphics in non-linear optimization is then evaluated from the points of view of model formulation and solution, and the requirements of an interactive system for realizing this potential are outlined.

Such a system is developed and described, together with a full account of its applications to both real and standard test problems. A novel application is the direct optimization of N -dimensional ($N > 2$) problems by visual analysis of 1 and 2 dimensional sub-problems, following on the formal development of an algorithm for this approach. The interactive control of conventional methods through computer graphics is also featured.

On the basis of these accounts an evaluation is given of the contribution which interactive computer graphics is capable of making in both teaching and research in the general field of non-linear optimization.

ACKNOWLEDGEMENT	(i)
ABSTRACT	(ii)
<u>CHAPTER ONE: Interactive Computer Graphics</u>	1
1.1 The Evolution of Computer Graphics	1
1.2 The Elements of Interactive Computer Graphics	4
1.3 Applications of Interactive Computer Graphics	14
1.4 The Design of Interactive Computer Graphics Systems	17
1.5 Summary	20
<u>CHAPTER TWO: Non-linear Optimization</u>	22
2.1 Preliminary Aspects of Optimization	22
2.2 Mathematical Models	23
2.3 Local and Global Optima	24
2.4 Classification of Objective Functions	25
2.5 Methods of Optimization	25
2.6 Comparison of Algorithms	37
<u>CHAPTER THREE: Interactive Computer Graphics in Optimization</u>	39
3.1 Interactive Computer Graphics in Optimization	39
3.2 Evaluation of Potential	40
3.3 System Requirements	42
<u>CHAPTER FOUR: An Interactive Computer Graphics System for Optimization</u>	44
4.1 Hardware Mechanism	44
4.2 Software	45

<u>CHAPTER FIVE: Applications to Test Problems</u>	67
5.1 Problems in One-dimension	68
5.2 Problems in Two-dimensions	89
5.3 Problems in N-dimensions	106
5.4 Plane Analysis Method	121
<u>CHAPTER SIX: A Practical Application</u>	159
6.1 Molecular Vibrational Analysis	159
6.2 Minimization Problem	168
6.3 Methods of Solution	169
6.4 Comparison of Methods	177
<u>CHAPTER SEVEN: Appraisal and Conclusion</u>	181
APPENDIX	184
REFERENCES	186

Interactive Computer Graphics

In this chapter some of the general principles concerning the history, applications and design of interactive computer graphics systems are discussed.

The introductory section presents the origin and the development of graphical techniques. In the following section, the facets of hardware and software for interactive graphics are considered.

An account of some of the application areas of interactive graphics and the merits of graphical computing methods is given in the next section. Finally, in the last section of this introductory chapter, some specific factors relating to the design of an interactive graphics system are considered.

1.1 The Evolution of Computer Graphics

The need for increased calculating power to solve complex scientific problems gave rise to the invention of computers. Since their invention, over thirty years ago, computers have been closely linked to the developments of many scientific and engineering methods. Until the late 1940's computers were only capable of producing numerical results, using a high speed line-printer. However, it was found that these numerical results were not sufficient when dealing with empirical quantities and data exhibiting complex relational attributes. It was felt that some numerical results could be more meaningful if presented in graphical forms rather

than tabular forms. There are several reasons for this assertion.

- (i) Graphical illustrations are easier to understand, specially if the user is looking for comparison or trends in large files of output data. In subject areas such as statistics, computer-produced tabulations are used as raw data to produce graphical presentation of data. These graphs are useful aids to the detection of errors in large input data files.
- (ii) When presenting the output graphically, there is more flexibility in the showing of emphasis on key areas.
- (iii) The graphs are more compact, since many pages of numerical output can be contained in a few graphs.
- (iv) Data can be seen in two dimensions as opposed to a one dimensional list of figures. This is particularly useful for showing relationships between variables, allowing a better interpretation of the data.

It was only natural that there would soon be a demand for the computer to be able to communicate its results in a graphical form whenever appropriate. However, technology has not allowed this feature to develop at so rapid a rate as the manipulation of numerical data, and it was not until 1953, that the Benson-Lchner Corporation introduced a digital graph-plotter. Graph-plotters operate slowly when compared with general computer processing. They are largely used in the place of manual graph plotting, which means that the graphical operation is a passive one, and is thus usually referred to as non-interactive graphics. The graph-plotters are distinctly inefficient when a sequence of further computation

depends upon the analysis of the displayed graph, and this drawback has led to the development of specialised equipment which enables the graphing operation to be performed under interactive control.

Interactive computer graphics can be defined as a close interaction between the user and the computer through the use of a visual display, an input device, and a special computer language. The user is able to communicate with the computer and to receive a direct response from it. This two way conversation may be of a graphical or pictorial nature, and, as a result of previous programming, the computer can analyse the output, perform calculations, and almost instantaneously present a revised display for further analysis if required.

The pioneering work in the field of interactive computer graphics was carried out by the US Air Force. In the early 1950's a system called SAGE (1), Semi Automatic Ground Environment, was designed in order to help to protect the United States from surprise air attacks. SAGE became operational with one computer centre in 1958 and has now been used in many centres.

SAGE presented visual information on aircraft positions, and the air force personnel could communicate with the computer using a light pen. This system laid the foundation for further hardware and software developments in the field of interactive computer graphics.

In 1962, Sutherland and Johnson (2), announced the development of a system known as 'Sketchpad', at M.I.T.

In parallel with Sutherland's work on sketchpad, research was being carried out in industry mainly at General Motors (3), and Itek Laboratories (4).

In 1959, with a contribution made by IBM, General Motors produced DAC-1, 'Design Augmented by Computer', a computer aided design system for the automobile industry. The Itek Laboratories were also involved in developing an on line graphics system for lens design. This system was based on the laboratory's early research work on graph plotters.

The systems mentioned so far had two bothersome characteristics in common;

- (i) the hardware was expensive
- (ii) the software used a great amount of time.

By the early seventies, there had been further hardware developments leading to the improved interactive computer graphics systems which are being used by industry and research establishments today. These developments include the emergence of the storage tube, the micro-processor and satellite computers, aiming to provide more computing power at substantially reduced costs, and hence more effective computer graphics.

Modern systems are usually described in terms of their hardware configurations, and their software needs and capabilities. These will be briefly discussed in the following section.

1.2 The Elements of Interactive Computer Graphics

This section briefly describes the hardware and software currently available generally for use in interactive graphics systems.

1.2.1 Interactive Graphics Systems

There are two main types of interactive graphics installation in common use -

- (a) Stand-alone system
- (b) Time-sharing system

(a) Stand-alone System

A typical system of this type would consist of a processor with 8k words, a visual display, and a graphical input-output device. The software of such a system is compact, yet fairly complex, frequently written in assembly language or even machine code.

This type of system can be used for applications which require little processing power. It offers the advantage of rapid personalised data processing, particularly when it possesses interactive features.

Many such systems exist, and perform useful jobs in industrial laboratories and research establishments.

(b) Time-sharing System

This type of graphics installation provides graphics terminals as part of a general time-sharing system. This system can be useful for applications which require the computer to perform extensive calculations under the control of the user. The processor in such a system is normally available for non-graphical applications, although it can support graphical applications.

The system used for the development of the project which forms the central part of this thesis is a time-sharing system. It consists of an IBM/370 main computer, linked to a Tektronix 4014 graphics terminal. The display file is held in the core of the main computer, where the display controller had access to it.

The main difficulty which arises with operating a time-sharing system is that each user receives services only intermittently. A user with a display however, requires that the display should be animated continuously, and that for certain services, the user should receive instantaneous response.

These conditions can best be met if the display file is driven by a small computer connected as a satellite to a time-sharing system. Such arrangement does offer several distinct advantages:

1. It provides an independent source of computing power.
2. Allows the flexibility of choosing the position of the equipment.
3. It can act as a concentrator for multiple display consoles.
4. It can operate as a stand-alone system without interrupting the host computer for servicing.

The main disadvantage of host-satellite systems is that experience in their use is relatively limited, and if the satellite is situated remotely from the host, program development can produce problems. Furthermore, if either the link or the host computer fails, the system is normally unuseable.

Thus, when there are doubts about the effectiveness of a satellite system, then either a stand-alone or a conventional time-sharing system with display unit can be considered to be better suited alternatives. However, continuing developments of graphics facilities allied to micro-computers may eventually swing the balance in favour of the stand-alone system for virtually all types of application, other than those involving major computations.

1.2.2 Hardware

An interactive computer graphics terminal is usually structured as an extension of a large computer. The basic elements of graphics hardware are -

Display Channel

Display Controller

Display Console

Input Devices

The function of this hardware is to allow application programs to use the display unit for graphical output, while running in the central processor unit.

A complicated picture will call for a long list of data and character descriptions. The graphical display instructions, the data containing the picture, or the result of an analysis can all be stored in the display file. The display file is continuously accessed by the display channels and graphical commands are sent to the display controller. This has two main advantages. First,

the display channels become much simpler devices, as the computer does not have to refresh them regularly. Second, the load on the computer can be reduced by increasing the flexibility of the display controller to handle some of the graphical instructions. This effectively changes the display controller into a small computer itself, called the display processor.

To produce a continuous picture the display processor reads the display file, and executes the operations required to display the picture, and refreshes it as often as fifty times a second.

Interactive computer graphics consoles require at least one input device for use in conjunction with the display. A keyboard is essential so that alphanumeric information can be entered, but it is also necessary to have some means of transferring geometrical information to the computer.

There are two ways for passing geometrical information to the computer,

- (i) pointing
- (ii) positioning

The light pen is one of the pointing devices which is more commonly used. It has the advantage over the other input devices in that its positional data is determined by the program, and it does not depend on any physical measurements of the actual pen position. The disadvantage of the light pen is that it is very sensitive to surrounding illumination, and it cannot be conveniently used with the storage tube.

The joystick, trackball and cursor are positioning devices, and they do use the screen directly. A more recent development in this field is the 'MOUSE' (5), produced by the Stanford Research Institute. All these devices operate on the principle of moving a control to pass two-dimensional information to the computer.

The input device used in the project described in this thesis is the cursor. Two crossed lines appear on the screen, by the user's request, and the co-ordinates of the point of interaction of the two lines can be recorded. The two lines can be repositioned by thumb-wheel when the co-ordinates of a new point are required.

1.2.3 Software

In a computer graphics system, the software produces an interface between the computer and the graphics devices. There are three fundamental components in the software -

- (i) The executive
 - (ii) The applications programs
 - (iii) Graphics software
- (i) The Executive

This part of the software attempts to blend the best features of the hardware, the applications programs and the user in such a way that the user can interact in a harmonious dialogue with the computer. The executives are usually supplied by the computer manufacturers, and most of them are

not graphics oriented, since they control all other programs running in a computer system. In a multi-terminal computer system, the executive is responsible for the sharing of time between all the terminals attached to the drive processor.

(ii) Applications Programs

The applications programs are similar to the problem-oriented programs in a batch system with the difference that they run in real-time. They specify what is to be displayed by producing definite commands or constructing a data structure representing the picture. The amount of structuring used by the applications programs to describe a picture depends entirely on the applications. If no particular structure exists, the picture will be described in terms of co-ordinates.

In general, a graphical system contains three groups of data -

- (a) the data base for the applications programs
- (b) the data structure representing the display
- (c) the data for the display file

In many systems the display file is combined with the display structure. Figure 1.1 shows the data path through a typical graphics system.

The commands in the applications programs are written in a high level language which must be able to program the applications, and to handle the graphics devices.

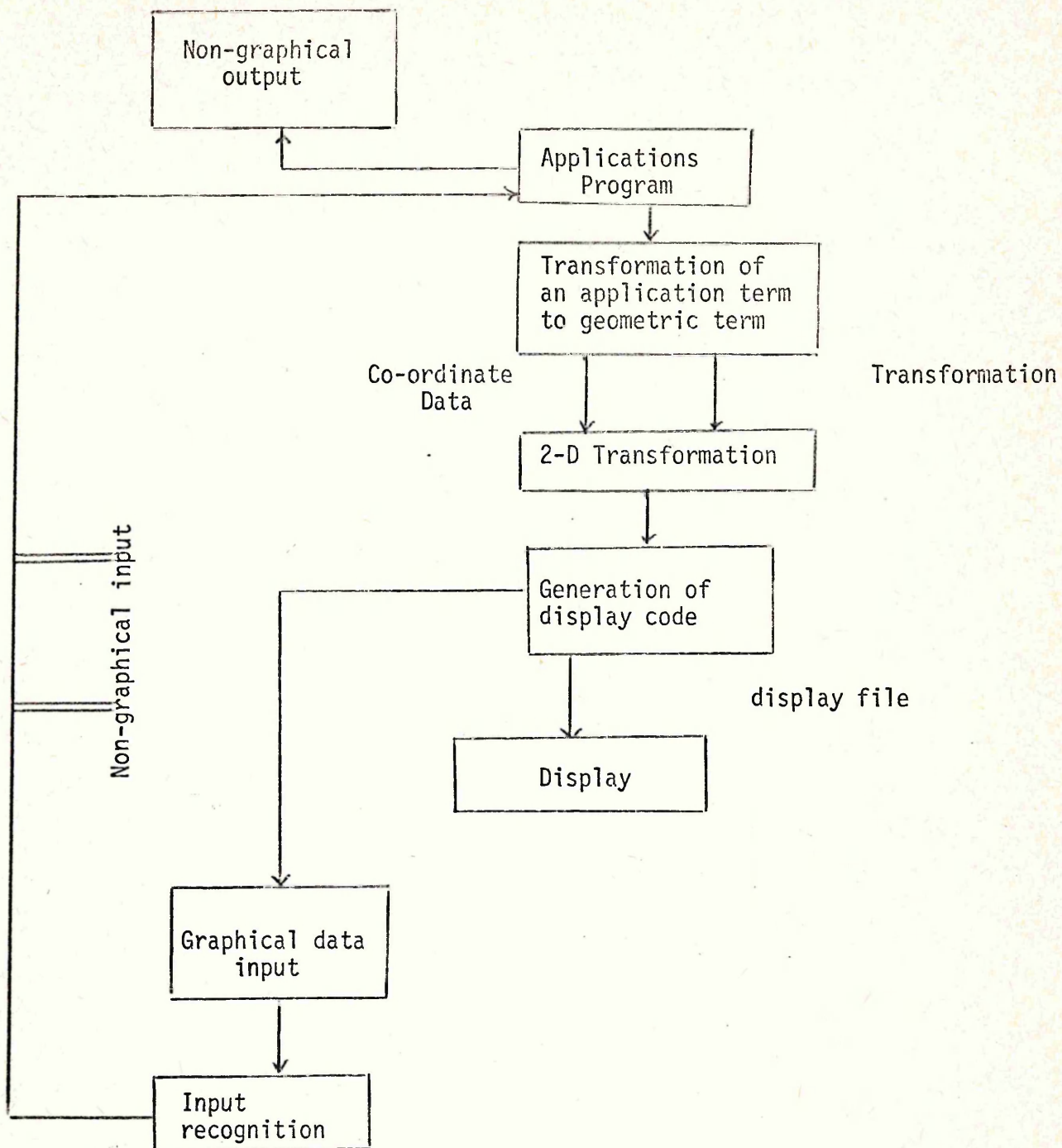


Fig. 1.1 Data Path Through A System.

FORTRAN is the most commonly used language in computer graphics. The main reason for this is its wide range of scientific applications, and also because on many small and medium size computers it is the only language used. Unfortunately, FORTRAN, has weaknesses. It gives no programming structure, and its subroutines and functions are non-recursive. The usage of its 'DO LOOP' and 'IF' statements is limited but it has the advantage of completing a task more rapidly than an equivalent program written in another high level language.

ALGOL 68, PL/1, BASIC and JOSS are some of the other high level languages used for computer graphics. The last two languages mentioned are conversational, but because of their limited data structure facilities are only suitable for informal problem solving. For the present time, however, APL remains the only conversational language suitable for complex problem solving using graphics.

A more recently developed language for computer graphics is EULER. This language is designed by Wirth (6), as a generalisation of ALGOL 60. The program structure facilities of this language are very powerful and it is a language of free type declaration.

In general, some of the high level languages have facilities for manipulating and structuring data, others have very flexible input-output procedure, and a few of these languages possess a good conversational mode. Recent

researches suggest that a language suitable for computer graphics users should have all the above mentioned attributes as well as being machine and problem independent. However, since a language with all these properties has not yet been introduced, FORTRAN remains the most widely used language in the general field of computer graphics.

(iii) Graphics Software

The graphics software consists of a large group of routines, which are called by the applications programs. This operation requires the use of one of the high level languages with an effectively large syntax. The languages are often referred to as graphics languages. One such graphics language is AED, Algol Extended for Design, developed at MIT (7).

The main function of graphics software is to transfer data between the applications programs and the display hardware.

The application-dependent part of the graphics software receives instructions from the applications programs to scan the data structure produced by the applications programs, and to generate a description in a two-dimensional space,

The display-dependent part of the software manipulates the two-dimensional field to a suitable form for the display hardware. Theoretically, the field extends to infinity in all directions, but practically it is limited to the range of the numbers presented.

In order to produce an efficient software the applications programs written in one of the high level languages are often combined with a graphics software package which can be used to deal with attention handling, display file routines etc.

The graphics software package used in the investigations described later in this thesis is known as TCS, Tektronix Control System. The package is a comprehensive set of subroutines which allows terminal-independent graphics programming. The design is basically system and computer independent and enables the experienced programmer to work at the terminal level, and it also provides the facilities for the occasional user to operate easily at the conceptual level.

1.3 Applications of Interactive Computer Graphics

The use of computer graphics techniques ranges throughout research, engineering, design and administration. Major application areas may involve several of these broad fields of activity and they can also overlap each other.

In general, interactive computer graphics can be applied to two main areas. These are -

- (a) Design, where emphasis is placed on drawing to assist the designer by stimulating the creative process.
- (b) Science, which requires less creative ability on the part of the user but requires ability to analyse the displayed information, in order to modify it as part of the problem-solving process.

1.3.1 (a) Graphics in design-oriented problems

Since 1960 efforts have been made to use graphics in many branches of industry to help reduce the cost and also to produce better results.

Developments in design have taken place in areas such as engineering, chemical design, textiles, animation etc.

Detailed descriptions of some of these applications are given by Green and Parslow (8). Numerous illustrated examples of design work in the aircraft industry in the 1960's are given by Prince (9). Recent work in this field has been carried out at MC Donnell Douglas Corporation who have found graphics valuable in building the new F-18 fighter plane.

One area which has profited from the developments in computer aided design is architecture. Paterson (10), has produced a system which improves the design and construction processes. This system replaced an earlier experimental program developed in 1965 (11), which has laid the foundation for further advances of graphics in architecture. One of the latest pieces of research work is being carried out at Leeds Polytechnic. There, the aim is to develop a computer architectural modelling system (12).

1.3.2 Graphics for scientific users

Although, computer graphics was introduced as a tool for designers, in recent years efforts have been made to use graphics for solving technical problems in scientific fields such as chemistry, physics, mathematics and medicine.

Scientists are usually concerned with rapid recognition of contours, trends, peaks and valleys which demonstrate inter-relationships between variables. To satisfy such needs attempts have been made to take advantages of progressive innovations in computer hardware and software that provides opportunities for generating information in graphical form.

Cardwell (13) describes certain practical applications of computer graphics at Oak Ridge National Laboratory, which have contributed to the development of nuclear reactors. Reactor development experiments generate large quantities of data which have to be analysed. Computer generated graphics have been found to materially aid the analysis by consolidating large volumes of line printer tabulated numerical data into geometric diagrams. The introduction of computer graphics has terminated the tedious manual plotting from voluminous line printer outputs. It has also allowed the parametric relationships to be deduced more rapidly and clearly.

Mac Elroy (14), also describes a computer model which was designed to investigate the conformation of molecules and subsequent complexes. The system is known as AIMS, Ames Interactive Molecular Modelling System, and it is used to study pre-biotic molecular evolution towards life. The system is capable of simulating molecular structures and their transformation. It comprises a library and four programs. Interaction, manipulation of molecular complexes containing a large amount of atoms, three dimensional viewing, and co-ordinate retrieval can provide the

biologist with a molecular modelling capability that is easier to handle and more reliable than the traditional wire modelling techniques. In addition, the system allows further investigation of structures by calculation of -

- (i) conformational energy
- (ii) the interaction between molecules or submolecular fragments

The above examples are only two of many examples of computer graphics in science. Cooper (15) discusses further the general principles involved in the design and application of interactive computer graphics systems to scientific problems, and quotes several additional examples.

The progress of computer graphics has not been as rapid as that in the design area, but it has been predicted that the development of more effective software, the eventual design of a good graphics language, and the emergence of substantially reduced cost and high performance hardware will widen the application areas of interactive computer graphics in science (16).

1.4 The Design of Interactive Computer Graphics Systems

The development of interactive computer graphics, depends to a great extent on effective system planning and design. Therefore, the art of design of an interactive computer graphics system is -

- (i) to determine, for any application which are the parts which are best treated by suitable interaction with the computer.
- (ii) to translate these parts into a suitable form of communication with the computer.

- (iii) to bring together, considering the time-cost limitations, the correct hardware and software which can process one or more particular applications.

However, in general, a system capable of processing the requirements of one application may not be suitable for another. Therefore, the overall aims of the graphics system should be considered alongside the hardware and software components.

Newman and Sproul (17), have identified three types of interactive graphics system, each designed to meet the need of a different type of user. They are -

- (I) Picture-editing systems
- (II) Specialised application systems
- (III) General purpose systems

Having identified the type of system to be developed, consideration should be given to -

- (a) interactive dialogue
- (b) command and control

(a) Man-machine Dialogue

It is essential to keep the user in mind when deciding on the mode of the interactive dialogue.

For the user, who on many occasions will be a different person than the programmer, the dialogue should display concise messages whenever possible, explaining what the user should do next.

The dialogue should be simple and consistent. Therefore, a single style of command should be chosen and used throughout.

This has the advantage that the dialogue can be designed so that abbreviated commands are accepted. Another alternative is by pointing or positioning. This type of command is machine initiated, since a menu of available options appears automatically at each decision point, and the user must select the required menu item.

Often, the decision about the choice of a type of dialogue will have to be made partly on economic grounds. However, to make the dialogue suitable for particular applications the best features of each basic form of dialogue may be combined.

(b) Command and Control

It is good programming technique to give the user as much control over the program being run, as the circumstances permit. Thus it is desirable that the user should be able to -

- (i) select any feasible sequence of operation, and
- (ii) return easily to earlier stages of the program to modify, if desired,
- (iii) to examine previously entered data.

All these require a degree of flexibility often not found in many existing graphics programs, which impose too many restrictions on what the user is allowed to do.

The user should be given control over the flow of the program using the keyboard or the light pen as the major control device over program options.

At all times the user should be made fully aware of the current position of the program, and of the choices of the functions available. The right amount of information should be provided, neither too much information to cause clutter, nor too little to leave the user in doubt. Therefore it is practical to employ a hierarchy of information messages ranging from a bare minimum amount of information for the experienced user, to optional more detailed messages for users who require guidance on any particular option. An additional feature is to provide messages to be printed in response to the command.

Finally, a user's error should be recoverable, and to help avoid errors it is advisable to include a procedure which must be followed by way of confirming a decision action. For example, when using a menu type of dialogue, all others but the selected item may be deleted, before a further action by the user causes the command to be obeyed.

1.5 Summary

The description of interactive computer graphics and its applications in the previous sections leads to the following observations:

- (i) Graphs and diagrams can be produced far more quickly and accurately compared with traditional methods
- (ii) When applied to scientific problems, the interactive graphics approach enables the scientist potentially to obtain a better understanding of the problems than can be gained by conventional methods.

- (iii) A graphical presentation of the relationship between two variables is immediately available, and features such as maxima, minima, trends, etc., become apparent more rapidly.
- (iv) Method deficiencies, software errors and system design faults, where they exist, quickly become apparent when using interactive graphics.
- (v) Through repeated interactive computer runs, it becomes possible to obtain a better understanding of the solution procedure. This is particularly so when the complete problem is approached by the use of a graphics system.

Therefore, it may be concluded that computer graphics is undoubtedly capable of making substantial contributions to the several phases of scientific investigations. Such contributions can be expected to be most significant in those investigations requiring human interaction in computer programs, and where the current state of the computation can be summarised graphically. A problem of this nature is described later in this thesis.

Non-linear Optimization

In this chapter the general aspects concerning optimization and some of the available methods are discussed.

The introductory sections present some of the preliminary concepts of optimization, and a classification of objective functions.

Brief accounts of some of the available methods, their development and their applicability are given in the remaining sections of the chapter.

The purpose of this chapter is to lay the theoretical foundations for later descriptions of interactive computer graphics used in the testing, evaluation and application of optimization methods.

2.1 Preliminary Aspects of Optimization

The requirement for methods of optimization may be said to arise from the mathematical complexity necessary to describe the theory of systems. Optimization methods are used to explore the local region of operation and predict the way that the system's parameters should be adjusted to optimize system performance.

In an industrial process, the criterion for optimum operation is often in the form of minimum cost, where the product cost can depend on a large number of inter-related control parameters.

In the scientific field, the performance criterion could be to minimize the integral of a squared difference. In both cases it is required to minimize a single quantity by manipulation of a number of variables.

Since the beginning of this century many solution methods have

been developed to solve linear and non-linear optimization problems. These methods aim at extremisation of some entity while satisfying side conditions known as constraints.

Linear programming methods are concerned with optimization of a linear function subject to linear constraints and their main feature is that they always lead to global optimal solutions for well formulated linear problems. In contrast to linear programming, however, only special types of non-linear problems can be solved with an assurance of reaching the global optimum. Depending on the non-linear problem, and the starting values assigned to the variables, a given non-linear algorithm may converge to any one of several different local optimal solutions.

In practice, most problems are constraint bound, but this type of problem can often be reduced to an unconstrained one by a suitable transformation of variables. In consequence, this thesis is mainly concerned with methods for unconstrained problems, some of which are described in this chapter and applied later on. However, the potential of interactive computer graphics in the solution of constrained problems is demonstrated in the last two chapters.

2.2 Mathematical Models

In general the objective function, F , depends on n real independent component variables, x_1, x_2, \dots, x_n , often assembled for abbreviation into an n -component column vector, or point, X .

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = (x_1, x_2, \dots, x_n)^T$$

In many practical optimization problems there are constraints on the values of the parameters which restrict the search region.

An objective function may be subjected to two types of constraints;

(a) Explicit constraints, which are defined as follows:

$$L_i \leq x_i \leq U_i \quad i = 1, \dots, N$$

where L_i and U_i are the lower and upper bounds on the independent variables x_1, \dots, x_N .

(b) Implicit constraints, which are expressed as:

$$A_j \leq F_j(X) \leq B_j \quad j = 1, \dots, N$$

A_j and B_j are the lower and upper bounds of the implicit constraint function F_j .

2.3 Local and Global Optima

As was indicated in section 2.1, an important part of non-linear optimization is the concept of local and global optima.

The parameters X_{opt} which give an optimum value F_{opt} of the objective function within a local area of search is termed a local optimum. The global optimum of a problem is the local optimum of all possible optima.

In practice, especially for non-linear functions, it is very difficult to determine if the optimum obtained is a global one or not. Normally, a point will be accepted if its position approximates to the expected location of the optimum as determined by the nature of the problem. Otherwise, further searches from a different initial point will need to be initiated.

Most optimization problems tend to aim at minimization of the objective function. Since maximization and minimization are equivalent problems, i.e. the minimum of (F) and the maximum of $(-F)$ occur at the same point, $\text{Min } (F) = \text{Max } (-F)$, only minimization is considered here.

2.4 Classification of Objective Functions

Objective functions can be grouped into four classes depending on whether they are finite or continuous, and whether their first or second order partial derivatives are available or not. The four classes are:

- (i) Objective functions with a finite number of discontinuities.
- (ii) Continuous objective functions.
- (iii) Objective functions which possess finite and continuous first partial derivatives.
- (iv) Objective functions which possess finite and continuous first and second partial derivatives.

The above classification gives an insight to the applicability of the available methods of optimization.

An objective function with two variables can be envisaged as the surface of a hilly landscape. The problem of minimization is to locate the lowest point on this surface with constraints defining the bounds within which the search is to be made. If the function is assumed to have a continuous and finite first derivative there will be at least one minimum in the feasible region. If there is no feasible minimum the required minimum must lie on the constraints.

2.5 Methods of Optimization

A non-linear problem can be approached by two sets of methods;

- (i) Analytical methods.
- (ii) Numerical methods.

Figure 2.1 shows a genetic tree of the methods which will be described in the following sections.

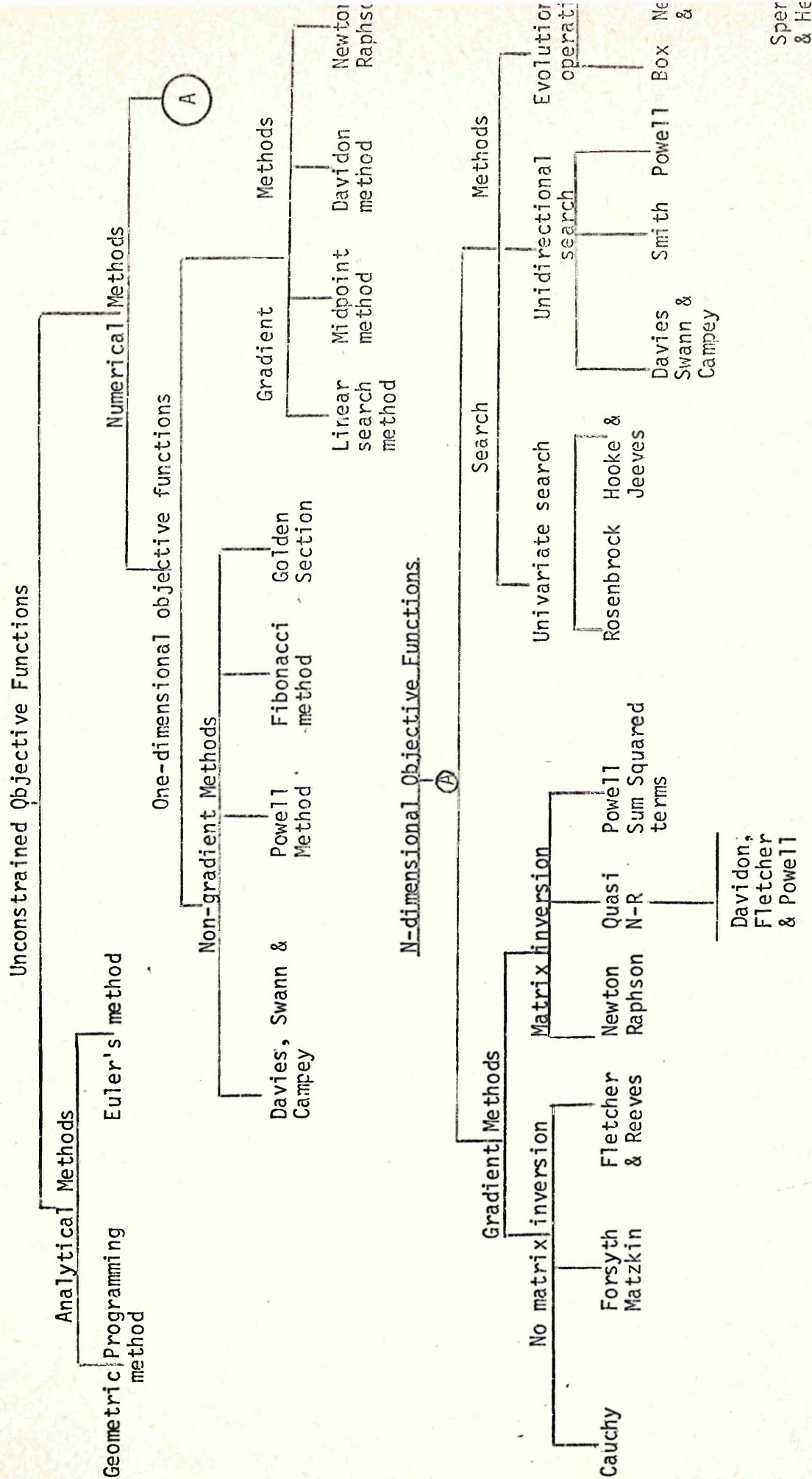


Fig. 2.1 General Tree of Optimization Methods.

2.5.1 Analytical Methods

In principle, these methods allow direct location of the optimum point. A variety of analytical methods exists, and two of these are Euler's method and Geometric Programming method (18). Both of these methods are only applicable to functions with finite and continuous partial derivatives. They also require information regarding the second partial derivatives. The complexity and size of modern problems limit the usefulness of analytical methods such as these.

2.5.2 Numerical Methods

Analytical methods present a computational problem in solving the large number of non-linear simultaneous algebraic equations which results from system description. The introduction of digital computers in the 1950's allowed optimization problems with more than three or four variables to be attempted, and this has led to the development of several efficient numerical methods.

Numerical methods are classified according to whether the function is

- (i) One-dimensional or,
- (ii) Multi-dimensional.

Some of the one dimensional methods have been extended to deal with multi-dimensional problems, either by multiple application or by generalising the one dimensional search in a multi-dimensional space.

2.5.2.1 One-dimensional objective function methods

These methods are important as some of them provide the basis of several multi-dimensional methods. They have two broad classifications:

(a) Non-graphical methods

These methods do not require explicit use of the first derivatives of the objective function. Nevertheless, they do require the objective function to have a finite and continuous first derivative. Furthermore, they require an initial point in the neighbourhood of the minimum, and some use repeated quadratic fits through the test points to predict the minimum itself. The quadratic fit limits the methods available to those functions with finite and continuous first partial derivatives. There are two commonly used algorithms, differing mainly in the means of selecting the three test points for a quadratic fit.

Davies, Swann and Campey's (19) method involves taking steps of increasing size from the given starting point until the function values at each step indicate that the minimum has been overshoot. An additional function evaluation at the centre of the final interval produces four equally spaced points which span the minimum. One of these points is redundant for the purpose of quadratic approximation. Therefore the point most remote from the evaluation giving a minimum value of the function is discarded. A single quadratic fit through the final three test points is then used to interpolate the minimum. The process is repeated until the desired degree of accuracy has been reached.

Powell's algorithm (20) differs from that of Davies, Swann and Campey in that a step of fixed length is taken from the given initial point, giving a second point. From the values of the function at these two points a third point is calculated. The minimum of a quadratic fit through these three points is

then located and the test point which yields the highest functions value is rejected. The two remaining test points and the point of minimum given by the quadratic fit are then taken to be the updated three points for a further quadratic fit. The process is repeated until the required minimum has been reached to within the desired accuracy.

Less efficient, but often used on account of their simplicity are the 'direct search' univariate methods such as Fibonacci (21) and Golden section (22) algorithms.

(b) Gradient Methods

There are two types of gradient methods. Methods which require the first derivatives of the objective function only, and those methods which need the first and second derivatives of the objective function in order to obtain the optimum.

The three methods of the first category which are investigated later in this thesis are,

- (1) Midpoint method
- (2) Linear search method
- (3) Davidson's method

The midpoint and linear search methods find the derivative of the objective function, and evaluate it at the two points defining the search region. The value of the derivatives at a third point is then calculated. This point replaces one of the other two points, depending on the sign of the derivative value when compared with the value of the other two derivatives.

As for the non-gradient methods, Davidson's method predicts the location of the minimum through repeated use of a low order polynomial fit (cubic) through the test points. This method is

very powerful, but is limited to functions with finite and continuous first and second derivatives.

The second type of gradient method consists of those which require explicitly the second derivatives of the objective function. One of the methods in this category is the Newton-Raphson method.

Some of the methods for one-dimensional objective functions mentioned so far have been extended to the multi-dimensional case where a sequence of one-dimensional searches is carried out in the multi-dimensional space along multi-dimensional search directions.

2.5.2.2 Methods for Multi-dimensional Objective Function

In the previous section the problem of finding the optimum value of a function with one variable was considered. It was found that the various types of methods were each most suitable for differing classes of functions, and situations. The same conclusion can be expected to hold when the number of variables is increased.

In general the optimization methods to solve multi-dimensional functions can be classified into two broad categories:

- (i) Gradient methods
- (ii) Direct search methods.

(i) Gradient Methods

The gradient methods can be divided into two groups, depending on whether they require matrix inversion or not.

(a) Methods with matrix inversion

The two methods included in this category are the Newton-Raphson method (23), and its variations, and Powell's algorithm (24) specifically for minimizing a sum of squared terms.

The Newton-Raphson method requires the exact determination of the inverse of the second derivative matrix, G . The iterative form is

$$x_{(k+1)} = x_k - G_{(k)}^{-1} g_{(k)},$$

where g is the -1 gradient vector. This method does not reach an optimum if the second derivative matrix is not positive definite. This also causes the method not to distinguish between maxima, minima and saddle points.

Many variations of Newton-Raphson have been proposed, and the most common ones are the Quasi-Newton or variate metric formulae. These were first proposed by Davidon in 1959 and were based on the fact that for a quadratic function

$$y = G d \text{ for all } k,$$

$$\text{where } y = g^{(k+1)} - g^{(k)}, \quad (2.1)$$

$$\text{and } d = x^{(k+1)} - x^{(k)}$$

Given a positive definite matrix $H^{(0)}$, the basic idea is to construct a sequence of matrices $H^{(k)}$ having the property that

$$H^{(k)} \rightarrow G^{-1},$$

by imposing the quasi-Newton conditions

$$H^{(k+1)} y = d \quad (2.2)$$

Biggs and Dixon (25) suggested an algorithm in which the Newton-Raphson prediction is made whenever it is possible to do so, but a step in the direction of the gradient is substituted for it where G is singular.

Powell's algorithm has proved to be efficient on a large range of functions. With this method the number of function evaluations required is usually for less than with methods which do not explicitly recognise the sum of squared terms, and also there is a reduction in computing from a third order to second order arithmetic step per iteration. An improved version of this method due to Powell (26) is available, which retains the advantages described above, and it also improves a constraint in the direction of steepest descent as the step length is decreased. It appears that for difficult test cases, this is one of the consistently successful algorithms in this category. However, as with all algorithms that involve the inversion of a matrix, its efficiency will fall as the number of variables increases.

(b) Methods without matrix inversion

One of the oldest of all numerical methods is Cauchy's steepest descent, which dates back to 1817. This method requires only a numerical approximation to the gradient, and the direction taken at each stage is that of the gradient. This method does ultimately converge, but the rate of convergence is rather poor.

Forsyth and Matzkin (27) advocate a procedure which involves steepest descent coupled with pattern move, known as the accelerated step descent search method. The advantages of this method are that it,

- (1) requires few iterations,
- (2) does not require evaluation of the gradient at the start of each one dimensional search,

(3) possesses the quadratic convergence property.

In the region around their optima functions can be approximated by a quadratic function of the form

$$F(X) = C + A^T X + \frac{1}{2} X^T G X \quad (2.3)$$

where G is an n by n positive definite matrix, A is a column vector of coefficients, and C is a constant.

A set of column vectors U_1, U_2, \dots, U_n can be defined that are mutually conjugate with respect to G , i.e.

$$U_i^T G U_j = 0 \text{ for } i \neq j.$$

Optimization methods based upon such vectors as sets of conjugate directions search, each direction in turn in search of the minimum. For a quadratic function of N variables, the optimum value of the function is expected to be found within N unidirectional searches. One such method which has been considered in this thesis is that of Fletcher-Reeves (28). In this method a linear search is undertaken along each conjugate search direction using cubic interpolation. This method is described in more detail in chapter five.

(ii) Direct Search Methods

In many problems, the function value is difficult to evaluate, and the expression for the gradient may be complicated or even unavailable. In both cases it is acceptable in principle to estimate the value of the gradient by numerical approximation methods and then apply one of the gradient methods. The calculation of these estimates may involve considerable computation, and difficulties often arise when numerical rounding errors affect the estimate of the gradient and interact with the con-

vergence criteria. Therefore, algorithms have been written that do not involve the gradient either accurately or approximately. These algorithms can be divided into three main categories:

- (a) Univariate search methods
- (b) Unidirectional search methods
- (c) Methods based upon evolutionary operations.

(a) Univariate search methods

These methods search along predetermined directions. Two of the more commonly used algorithms in this category are those suggested by Hooke and Jeeves (29) and Rosenbrock (30).

The Hooke and Jeeves method is applicable to any continuous objective function. It combines univariate search with pattern moves. The basic pattern move takes incremental steps after suitable directions have been found by univariate search. If the search progresses well in terms of decrementing the objective function, the step size is then increased, otherwise the step size is decreased. When the step size is reduced below a set value the search is terminated.

The Rosenbrock algorithm is similar to the Hooke and Jeeves method in following a 'valley', but it requires the re-orientation of the orthogonal direction to do so. The re-orientation of direction vectors is achieved by using the Gram Schmidt procedure. This algorithm does not depend on the properties of quadratic functions, and it usually obtains a local optimum. However, for functions which are quadratic near their optimum this method is not efficient when compared with other methods which take the quadratic convergence property into consideration.

(b) Unidirectional search methods

There are two variants of the unidirectional search approach, namely that of Davies, Swann and Campey and further methods based on conjugate directions.

The Davies, Swann and Campey method can be considered as a development of Rosenbrock's proposal of rotating the search directions. The final stage of linear search consists of a quadratic interpolation which could equally be replaced by that of Powell. This method is restricted to functions with finite and continuous first partial derivatives, because of the quadratic interpolation used during the linear searches.

The methods of conjugate search direction use a sequence of one-dimensional searches along conjugate search directions and have the advantage of being quadratically convergent.

One of the first methods to be based on conjugate search directions is due to Smith (31). It has the disadvantage of being rather inefficient for problems involving a large number of variables. There are two reasons for this inefficiency. First, the method is unable to continually update the test point so that the optimum can be found to the required degree of accuracy for a non-quadratic objective function. Second, the first of the N search directions is explored N times more frequently than the N th search direction.

Powell's method (32) is an improvement on Smith's method and uses a different approach for generating the conjugate directions. This method may fail to produce an optimum for functions which have a steep valley skewed to the co-ordinate

directions. This may be due to either loss of the quadratic convergence property, or by failure to generate new conjugate directions.

(c) Evolutionary operation methods

The original idea of evolutionary operation for optimization was introduced by Box (33) for a large scale industrial operation using statistical methods. Although the method demands only continuity of the objective function, it has the disadvantage that a large number of function evaluations are required. If N is the number of independent variables of the objective function, then the method requires $2^{(N+1)}$ function evaluations.

The use of a simplex in a 'hill climbing' context was first suggested by Spendley and Hext (34). The number of vertices used in the simplex method is $(N+1)$ of which N points are re-used in the next iteration. Hence, only one function evaluation is required for each iteration. The original simplex method of Spendly and Hext had difficulty in negotiating a narrow curved valley and once out of the valley the simplex method could not expand for a quick move towards the extremum point. The method was further improved by Nelder and Mead (35) through the use of reflection, expansion and contraction coefficients. The main advantages of this method are; first, it does not involve unidirectional searches, and second, it does not use quadratic approximations to the function, so it can therefore be applied to functions that contain discontinuous derivatives and to functions where the neighbourhood of the optimum may not be quadratic.

The major defect of this method is that it produces a good approximate answer efficiently but then terminates without locating the exact optimum.

2.6 The Comparison of Algorithms

In the previous sections some of the different methods of finding the optimum of a function of one and N variables have been discussed. It can be concluded that there is no one algorithm which could be considered as the most suitable for all types of optimization problems. Some of the methods may work for certain objective functions, whereas they may be very inefficient for others. However, the choice of methods depends entirely on the type of objective function to be optimized. For example, if the objective function does not have the sum of squared form, and its first derivatives are not available, then it is advisable to use a direct search routine. On the other hand, if the objective function is likely to contain discontinuities in its gradient, then the simplex method is recommendable. For functions with a large number of variables, the Fletcher and Reeves method may be used since it does not require any matrix to be stored or inverted. If all the requirements are available, then a Quasi-Newton method can be applied to produce a rapid optimum value.

Ghani and Barnes (36) give an account of some of the available methods, modifications to them and their applicability to different classes of functions.

Box (37), also compares the performance of eight methods for unconstrained optimization using a set of problems with upto twenty variables.

In the later chapters of this thesis some of these methods are described in more detail and their performances are illustrated graphically for comparison purposes, using a set of test functions.

CHAPTER THREE

In the first chapter some general aspects of interactive computer graphics and its uses were described. Chapter two introduced the basic optimization problem and discussed solution methods.

The present chapter aims to bring together the themes of the first two chapters and to assess the contribution which interactive computer graphics might make in the field of non-linear optimization.

The introductory section gives a brief account of the potential of interactive computer graphics in optimization, in problem formulation, model development and solution, presentation of results and in teaching and research.

In the following section some of the advantages and disadvantages are discussed in an evaluation of this potential.

Finally some brief requirements of a system to realise the potential benefits are given.

3.1 Interactive Computer Graphics in Optimization

Non-graphical methods are well established in optimization and recent technological developments have enabled the computer to contribute directly to graphical work in this area.

Interactive computer graphics is undoubtedly capable of making substantial contribution to the many stages of optimization. Its real potential in situations where a system is available are:-

- (1) Problem formulation
- (2) Model development, which involves data analysis.

- (3) Solution of model by a choice of methods. This may be combined with sets of known methods. The user chooses a method and can participate in controlling the solution process, observe the results from different methods and use this information and experience to evaluate various methods. The results obtained by numerical methods could also be compared with the results of visual analysis in the case of one and two dimensional problems or sub-problems. This raises the interesting possibility of solving problems in more than two dimensions by visual analysis.
- (4) Presentation of results. The results of a problem, both intermediate and final, can be presented graphically as well as numerically. Graphical presentation of the solution gives the user a better understanding of the behaviour of methods, and interpretation of the results becomes easier. This can also lead to the development of new methods.
- (5) Evaluation of solutions. Using graphical display of solutions, comparisons can be made between the behaviour of model and that of the real situation.

3.2 Evaluation of Potential

Some of the general benefits and problems connected with the operation of interactive graphics system have been discussed in chapter one. Now some of the advantages and disadvantages associated with the use of interactive computer graphics in optimization are presented.

3.2.1 Advantages

The main advantages of interactive computer graphics in this context are:-

- (i) Faster input/output, enables decisions to be made and acted upon more quickly.
- (ii) Speedier extrapolation to optimal solutions is achieved through graphical presentation and analysis of intermediate solutions.
- (iii) Gives some insight into how the methods work for both two and N-dimensional problems. This is essential because it enables one to obtain a visual analysis where methods have failed to converge.
- (iv) Interactive computer graphics can also be useful to newcomers in the field of optimization. It gives them a 'feel' of the problem and a better understanding of how different optimization methods work. This is achieved by the user's intervention at several decision points which are included in the program.

3.2.2 Disadvantages

- (i) The main disadvantage of using interactive graphics is that it is heavy on function evaluation, a major difficulty with complex problems. However, the difficulty can be partially overcome by using lower definition graphics.
- (ii) Complex problems inevitably lead to slow picture generation. Applications are limited to situations where pictures can be produced in a reasonable time, or where relatively few are required.

- (iii) The combination of applications, optimization and graphics software may overload the computer system. Therefore economics need to be made whenever possible.

3.3 Systems Requirements

Many applications recently reported are either not interactive or not graphics based. In most areas of optimization where interactive computer graphics has been involved, its use has been restricted to either early presentation of the initial input data or the final analysis of the result. The complete research process has seldom been carried out through the medium of interactive graphics.

The quality of interaction can vary depending on the degree of involvement demanded of the user. An optimization program will be effective when it is able to provoke a response on the part of the user. Having initiated a response from the user, it is then essential to provide a feedback quickly. Without any feedback at all, the user will have no basis on which to modify the performance of a method, or to improve the input values. The combination of resources in a truly interactive graphical system is thus a powerful one, but it will be a useful tool when applied to certain optimization problems.

Before designing a graphics system a decision needs to be made as to whether the system is for non-expert or expert users, and whether it is designed for teaching or research. If a combination is required then special instructions should be provided to allow both types of users to decide on the degree of interaction and decision making in the optimization process. A new user can then

supply only the initial information to activate the process and allow other decisions to be taken by the system until he becomes more familiar with the system. This gives the user time to analyse the optimization pattern and also an insight into how the system works. For example a new user may not be able to choose a suitable step size to draw a curve, in which case the original step size may be chosen automatically by the system. Observing the curve the user can then reduce the step size to a desired value at the critical region.

In some cases, however, even an experienced user may not have sufficient information about a particular function. Therefore, it may be advisable to allow the system to provide the user with more information about the function, before participating in the optimization process. For example in cases where the functions are complex it may be difficult to choose the contour size, the user can allow the contour to be drawn by the system, and, by analysing the contour its size can be increased or reduced as required.

In chapter four one such system and its features have been described and its applications are illustrated in chapter five and six.

CHAPTER FOUR

An Interactive Computer Graphics

System for Optimization

In this chapter a system consisting essentially of two sets of comprehensive programs is described. The programs are written in FORTRAN and implemented on an IBM 370 computer with a Tektronix graphical display unit.

The chapter begins with a section describing the general aspects of the graphics system. This introductory section includes

- (i) The hardware mechanism.
- (ii) The programming language.

In the main section which follows, some general aspects of the software are discussed.

4.1.1 Hardware Mechanism

The Tektronix 4014-1 graphics display terminal is shown in Figure 4.1. The device is linked to an IBM 370, and it consists of a storage tube screen together with a keyboard. The Tektronix is connected to a hard-copying device, so that the user can make and retain copies of the graph-plotted outputs.

The user can interact with the display unit in two ways, by either,

- (a) the keyboard, which is a hardware facility enabling textual and numerical information to be entered, or
- (b) the cursor which is a software means of interacting with the display.

A track cross may be programmed to appear on the screen, then



Fig. 4.1 Tektronix Visual Display Unit.

the user can move the centre of the cross to a desired point and its co-ordinates are recorded. These co-ordinates, if acceptable to the user, can be returned to the program for further calculations, otherwise another point can be selected and examined.

4.1.2 Programming Language

An important point to be considered when writing programs for computer graphics is the choice of programming language. It was decided to use FORTRAN in this project. Some of the advantages and disadvantages of FORTRAN have already been discussed in chapter one, but the factors which dominated the choice here were;

- (i) The availability of the graphics package, TCS, which is written in FORTRAN.
- (ii) The scientific nature of problems which were to be used.
- (iii) Some of the existing FORTRAN subprograms for optimization could be incorporated.

4.2 Software

The system is designed for the user who is familiar with the fundamental methods and terminology of optimization. The user is also expected to have some knowledge of computer graphics. However, brief instructions are displayed to help the user at certain decision points. The variable names and messages appearing on the display are made compatible with optimization and graphics terminology which should readily be understood by the user.

The system produces graphical representations of functions, and allows the user to apply either a visual or an analytical method to obtain solutions to problems. The programs within the system can be divided into two broad categories,

- (i) Control programs
- (ii) Optimization programs

The optimization programs are used to locate the optimum in three types of problems -

- (a) Problems in one dimension
- (b) Problems in two dimensions
- (c) Problems in N-dimensions

4.2.1 Control Programs

These programs are used in the system to provide substantial interaction between the user and the display. They also control the presentation of graphical output by obeying user's commands. These programs consist of sets of sub-programs, which are -

- (i) Terminal control system
- (ii) Subroutine MENU
- (iii) Subroutine GRAF
- (iv) Subroutine FUNEQ
- (i) Terminal Control System (TCS)

This is a Tektronix software package and consists of a set of FORTRAN subroutines. The Terminal Control System subroutines communicate with one another primarily through the Terminal Statics Area, a set of common variables which continuously represents the current state of the terminal and maintains the data necessary to generate output according to the user's level of need.

These subroutines give many conveniences to the user. Bright and dark vectors are displayed on the terminal screen. The former indicates the 'DRAW' routine and the latter a 'MOVE' routine which is the equivalent of a bright vector. Calls to

other subroutines such as subroutines to provide the user with the cursor or to start the process on a clear screen are also dependent on TCS.

(ii) Subroutine MENU

The user interacts with the display by selecting the next stage in the program. At the end of each stage, the subroutine MENU (ICU) is automatically called to display the available options at that stage. The displayed options are identified by numbers or letters, corresponding to the instructions in the program. The instructions are obeyed when the options are selected and the appropriate keys of the keyboard are pressed. Subroutine MENU (ICU) has only one parameter, ICU, which acts as a code for the list of options to be displayed. The list of options to be displayed at different stages of the optimization process is controlled by the value assigned to ICU at various stages. Figure 4.2(a) and Figure 4.2(b) show two different sets of options displayed by the subroutine MENU (ICU) at two different stages of the program.

(iii) Subroutine GRAF

Prior to the graphical display of the function, the user may be requested to input the search area or the extreme ranges of the x and y axis. These extreme values determine the size of the plotted output, since they are representative of the size of the window within which the curve is plotted.

After each graphical display the user may wish to enlarge the displayed graph for closer observation and better analysis. This system was designed and tested on a CRT display unit, and

Fig. 4.2a

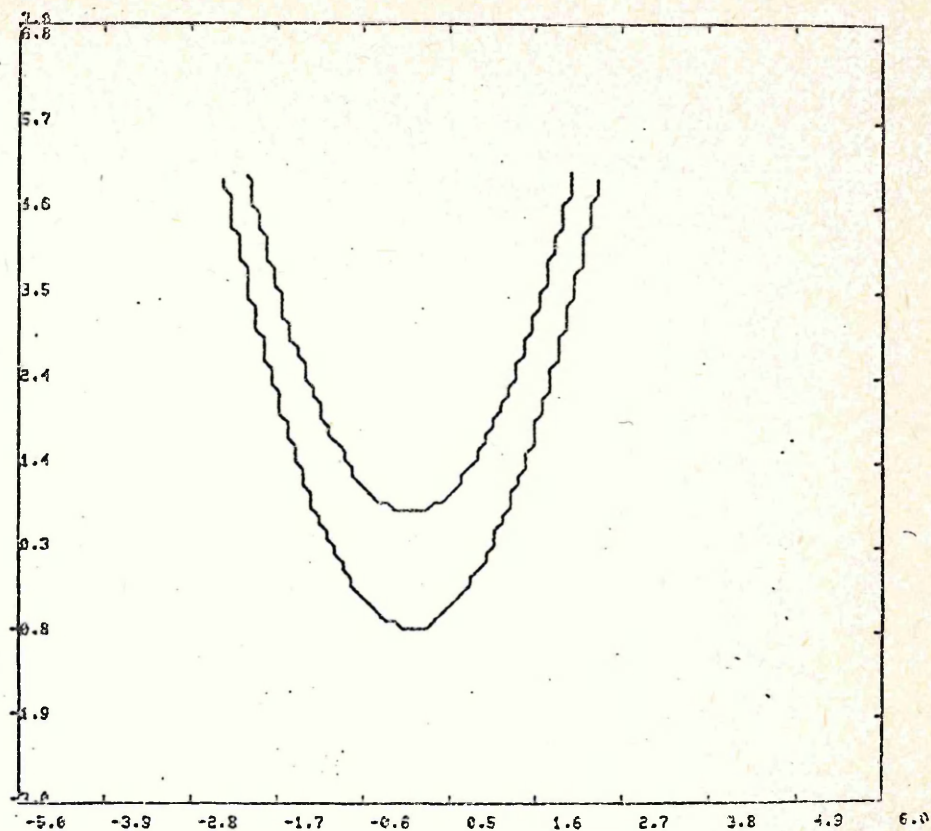


Fig. 4.2b

THIS SET OF PROGRAMS ARE USE TO FIND MIN. OF N-DIM FUNCTION
IT CONSIST OF FOUR ROUTINES ONE OF WHICH DRAWS FUNCT. CONTOUR
IF THE N-DIM. ROUTINES ARE NO LONGER IN USE , THEN TERMINATE
THE PROCESS AS AND ASK FOR A LIST OF ONE-DIM.ROUTINES
IN THE PROGRAM LINEAR CONSTRAINTS ARE DRAWN AS STRAIGHT
LINES

IN THE PROGRAM IF CONSTRAINTS OF IN THE NON-LINEAR FORM
ARE DRAWN AS DOTTED CONTOURS.

IF THE CONSTRAINTS ARE LINEAR, THEY WILL BE DRAWN AS
STRAIGHT LINES WITH THE LETTER (C), WRITTEN AT EACH END.

A LINEAR OBJECTIVE FUNCTION IS DRAWN AS A STRAIGHT LINE.

A NON-LINEAR OBJECTIVE FUNCTION IS DRAWN AS CLOSED CONTOUR.

WHEN THE PROBLEM HAS CONSTRAINTS THE SIGNS ARE GIVEN AS

A VALUE (-1), IS FOR (<) INEQUALITY SIGN

A VALUE (0), IS FOR (=), EQUALITY SIGN

A VALUE (+1), IS FOR (>), INEQUALITY SIGN

PRESS Y FOR YES, AND N FOR NO, IN THE PROGRAM

INTD, IDENTIFY THE METHOD TO BE USED

INTD-1, PROGRAM TO DRAW CONTOURS OF A FUNCTION

INTD-2, THE PLANE ANALYSIS ROUTINE

INTD-3, FIND MIN. OF FUNCT. BY FLETCHER-REEVES METHOD

INTD-4, FIND MIN. OF FUNCT. BY SIMPLEX METHOD

INTD-5, FIND MIN. OF FUNCT BY FLETCHER-POWELL METHOD.

NOW CHOOSE A METHOD AND PRESS THE APPROPRIATE KEY

Fig. 4.2 Possible Displays of Options.

so the zooming effect was not directly available. Therefore, to achieve zooming a new range of extreme points has to be specified. This is carried out by pressing the key representing the option 'New window'. This activates a call to the subroutine GRAF, and the user can then input the appropriate new values for a close-up of the output.

This subroutine can also produce two types of axis, specified by the user to suit the graphical output.

The subroutine GRAF is of the form

```
CALL GRAF (DXA,DXB,DYA,DYB,GXA,GXB,GYA,GYB,IGRF,IGA)
```

where the variables have the following meanings -

DXA,DXB,DYA,DYB - The range of the window in user's units. These are automatically adjusted to suitable values when the user declares the extreme points of the x and y axis.

GXA,GXB,GYA,GYB - The extreme points of the axes, which must be provided by the user.

IGRF - A code which controls the adjustments of the screen window. The value of IGRF changes automatically in the program when the modification has been made.

IGA - The value for this variable specifies the type of axis required.

IGA = 1 axes for one-dimensional functions.

IGA = 2 axes for two and N-dimensional functions.

The above values are automatically assigned to IGA, but the user may over-ride this and select the type of axis preferred.

Some illustrations of the capabilities of GRAF are given in figures 4.3(a) and 4.3(b).

UNREL. FOR X AND Y AXIS
1 2

FUNCTION VALUE IS 0.0000
AT POINT 1.0001 1.0001

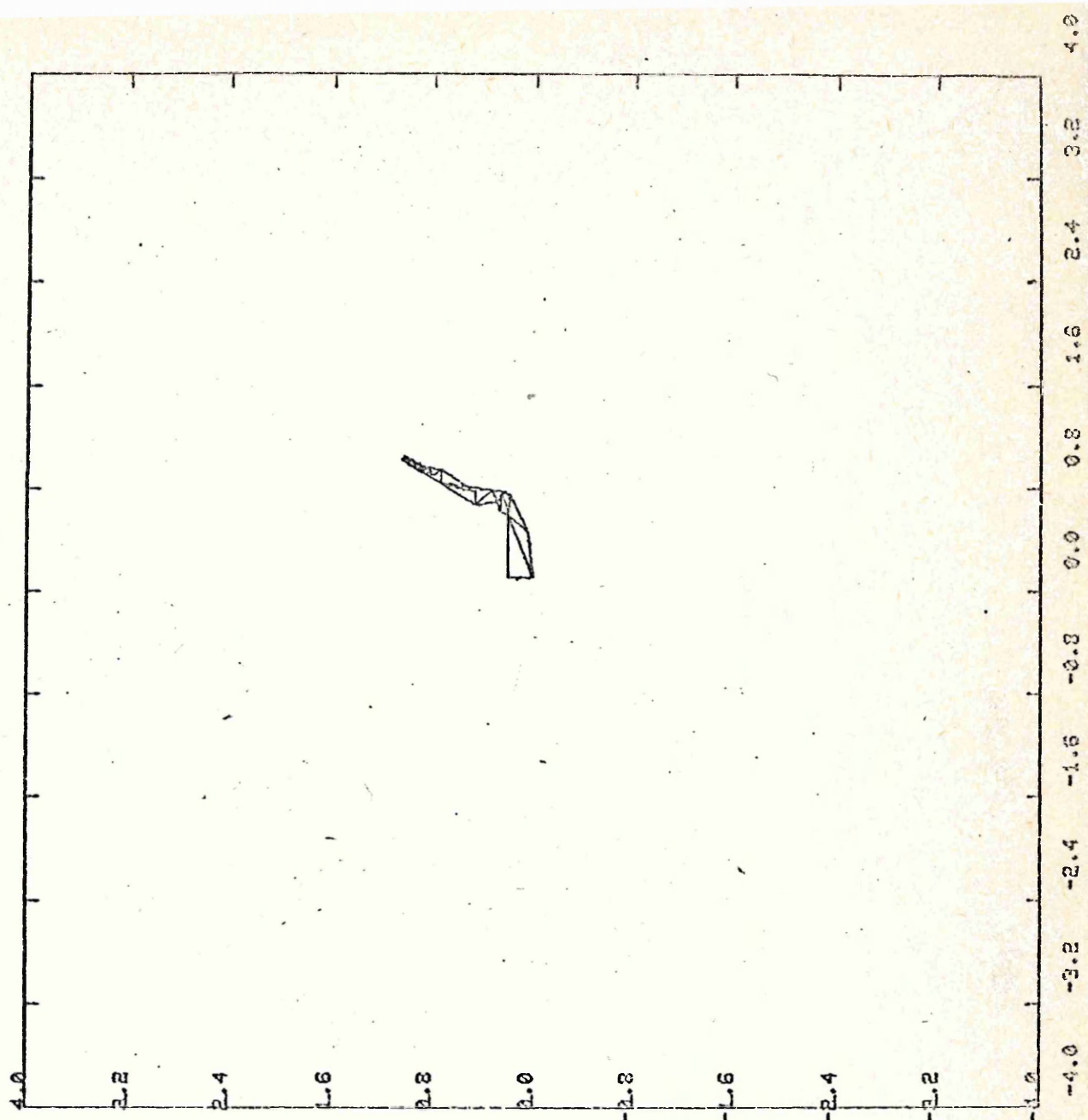


Fig. 4.3a Search Pattern of the Rosenbrock Function.

PRESS A TO FINISH
PRESS B FOR NEW INITL. PTS.
PRESS C TO CHANGE THE WINDOW

FUNCTION VALUE IS .0.0000
AT POINT 1.0001 1.0001

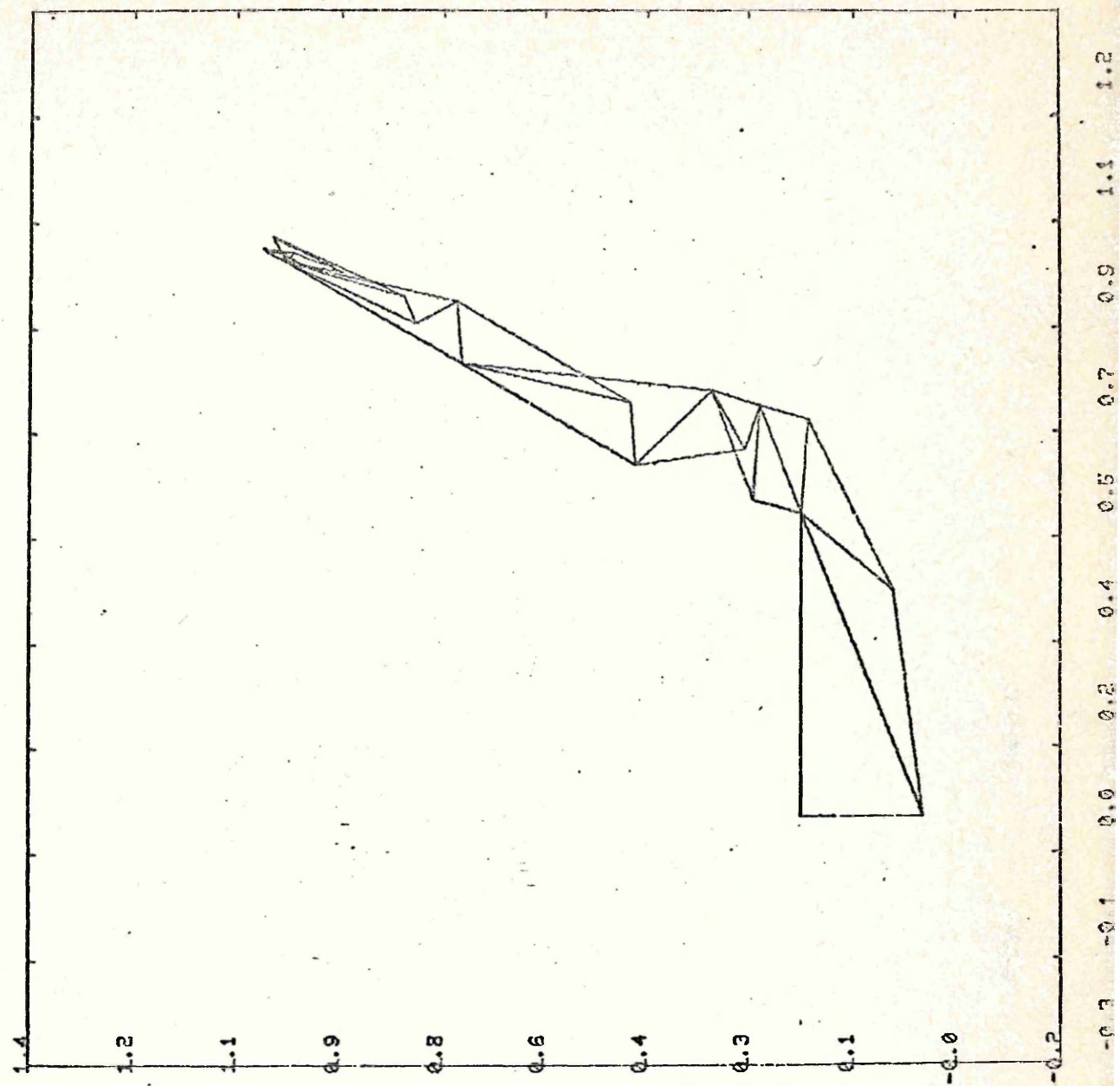


Fig. 4.3b

Display of Zooming Effect of the Search
Pattern of the Rosenbrock Function.

STOP
*END
*JOB CPU TIME=00M:52S
*GO

VALS. AFTER 10 EVALS.
F= -12.09 X= 2.544

VALS. AFTER 20 EVALS.
F= -34.94 X= 3.584

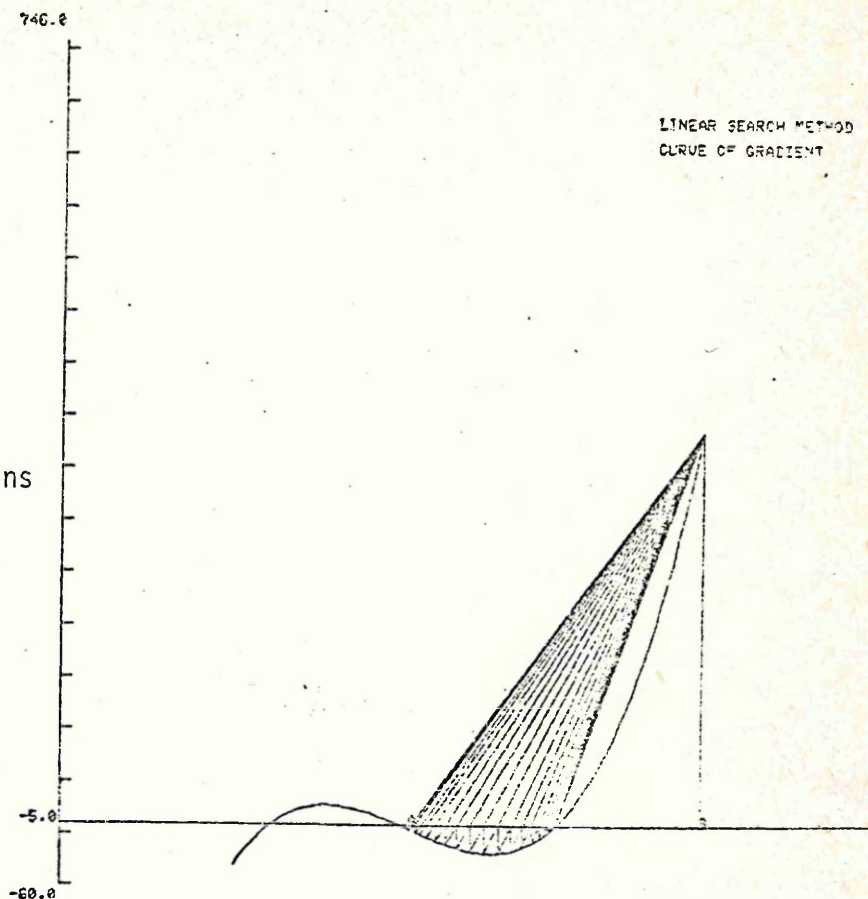
VALS. AFTER 30 EVALS.
F= -34.96 X= 3.529

MIN. REACHED AFTER 30 ITER.
F= -34.96 X= 3.529

LINEAR SEARCH METHOD
CURVE OF GRADIENT

Fig. 4.4a

Axes for One-
Dimensional Functions



PRESS X TO FINISH
PRESS Z FOR CURSOR
PRESS C FOR F.P. METHOD
PRESS D FOR MIN. COOR. FINDING
PRESS E,F FOR 1,2ND CONTOUR

PRESS G FOR NEW FUNCT.
PRESS H FOR NEW PAGE & METHOD
PRESS I STORE MIN. COOR. OF CONTOUR
PRESS J IF ONLY A NEW PAGE IS NEEDED
PRESS K TO CHANGE THE WINDOW.

PRESS L TO CHANGE THE STEP SIZE
PRESS ANY OTHER KEY FOR START
WITH CURSOR, A...I, AS BEFORE

THIS PROGRAM DOES NOT HAVE
ANY CONSTRAINTS

DO YOU WISH TO INPUT CONTOUR?
N

RANGE OF X, REAL
0.813 9.0 -9.0 0.313
RANGE OF Y, REAL
3.747 9.0 -9.0 3.747
VALUES CORRECT?
Y

RANGE OF X, REAL
1.985 9.0 -9.0 1.985
RANGE OF Y, REAL
2.177 9.0 -9.0 2.177
VALUES CORRECT?
Y

PT 1=	0.813	3.747	1.200	1.000	35.905
PT 1=	1.985	2.177	1.200	1.000	53.872
PT 1=	2.315	1.487	1.000	1.000	51.311

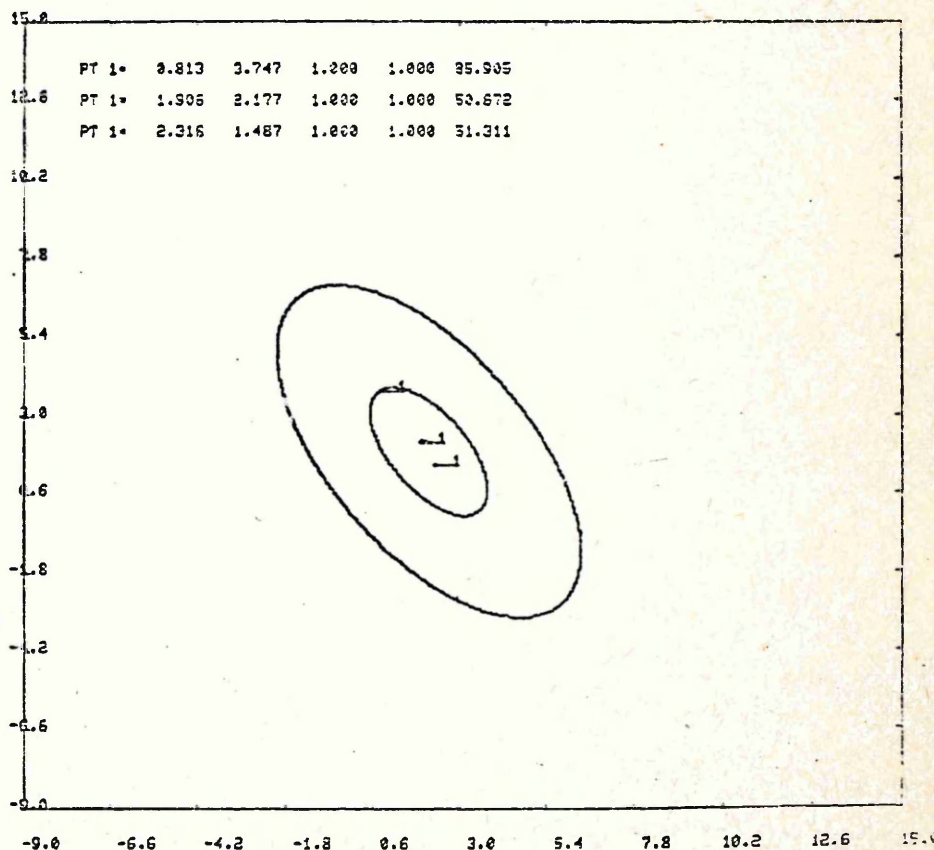


Fig. 4.4b

Axes for N-Dimensional
Functions.

FMIN= 294.000
FNC.EVAL. KN1= 1000

Figure 4.3(a) shows a search pattern using the Nelder and Mead method to minimize the Rosenbrock function,

$$F(x) = 100 (x_2 - x_1^2) + (1 - x_1)^2 . \quad (4.1)$$

Figure 4.3(b) shows a close-up of the above search pattern. This is obtained by a reduction in the size of the window.

Figures 4.4(a) and 4.4(b) show the two types of available axes, with IGA = 1 and IGA = 2 respectively.

(iv) Subroutine FUNEQ

Before choosing a method of optimization the user must identify the type of function which is to be optimized, i.e. a one-dimensional or an N-dimensional function. This information is used for the display of options. The user is then requested to set up the function which must be in polynomial form, and a FORTRAN call to the subroutine 'FUNEQ' allows the user to do so. The user must supply the subroutine with the polynomial coefficients in order of ascending power of the variables. From this information the derivative of the function is set up in the subroutine and its value is calculated whenever required. This facility affords considerable interactive flexibility for testing optimization methods on new or established polynomial test functions. The system is easily extended to cope with test functions other than those in polynomial form.

The subroutine 'FUNEQ' is automatically called at the start of the main program, so that the first function can be set up. But after each optimization process, the user is given the choice of either obtaining the optimum value of the same function with another method, or to set up a new function and evaluate its optimum value. Such an arrangement allows the

user to analyse and calculate the optimum value of more than one function in one session. This also saves computer time, since the user does not have to load the program each time a new function is optimized and, therefore, continuity of the operation is maintained.

4.2.2 Optimization programs

Once the function is identified and set up the user then selects one of the available optimization methods. Those presently available are discussed below.

(i) One-dimensional functions

For a one-dimensional function the user can choose one of four optimization methods. They are:

- (a) Midpoint
- (b) Linear search
- (c) Davies, Swann and Campey
- (d) Davidon's cubic search

The system is also flexible enough to allow additional search methods to be included and tested.

The optimum can also be obtained visually, before or after applying one of the above methods. The cursor is used to find the optimum value. The cursor locates the co-ordinates of a point on the displayed curve of the function. These co-ordinates correspond to the variable and the function value at that point. For comparison purposes the user can find the optimum value of a function by all four analytical methods as well as by visual means.

The user is required to specify whether the function is to be displayed after each method is terminated or not.

Figure 4.5 shows the minimum of the function

$$F(x) = x^4 - 4x^3 - 6x^2 + 4x + 4 \quad (4.2)$$

being located usually on the function curve. The value of the function is displayed on the left-hand side of the screen.

Figures 4.6(a), 4.6(b), 4.6(c) and 4.6(d) show the minimum of the function (4.2) found by the midpoint, linear search, Davies, Swann and Campey and Davidson's methods respectively.

(ii) N-dimensional objective functions

To find the optimum of an N-dimensional function three methods have been used. They are

- (a) Fletcher and Reeves method
- (b) Nelder and Mead, simplex method
- (c) Plane analysis method

Detailed descriptions of the use of each of these methods are given in the next chapter. However, the distinctive characteristic of method (c) is its use of two-dimensional slices in N-dimensional space.

Due to obvious limitations, contours of an N-dimensional function can only be shown as a series of two-dimensional constructs. The facility provided in the system allows the user to specify the two dimensional space for which the contours are to be drawn. Before entering in the optimization routine the user can produce a series of contours at different planes, record any desired values using the cursor, then start optimization with these values.

The program is also capable of displaying the contours of constrained functions as well as unconstrained ones.

Fig. 4.5

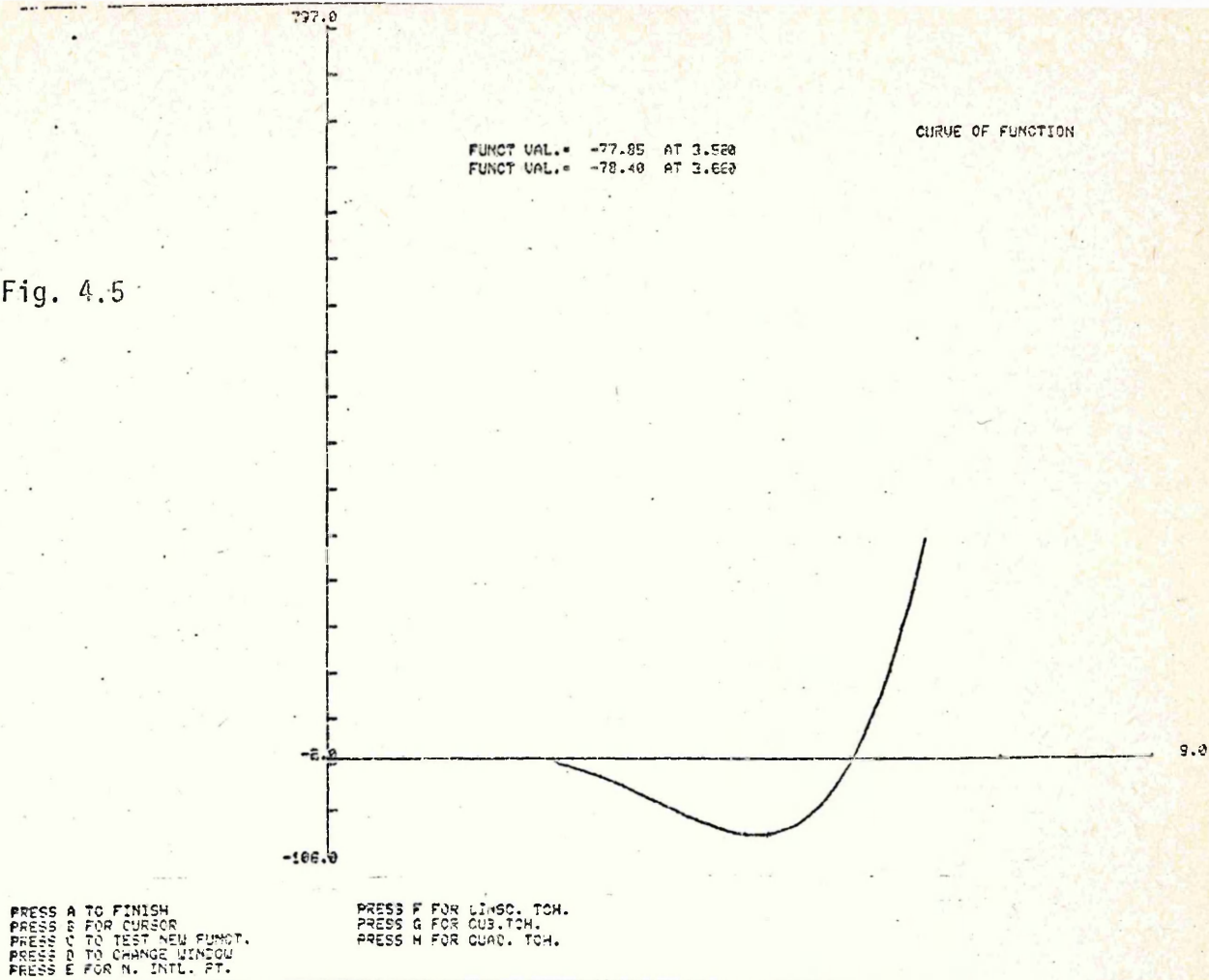
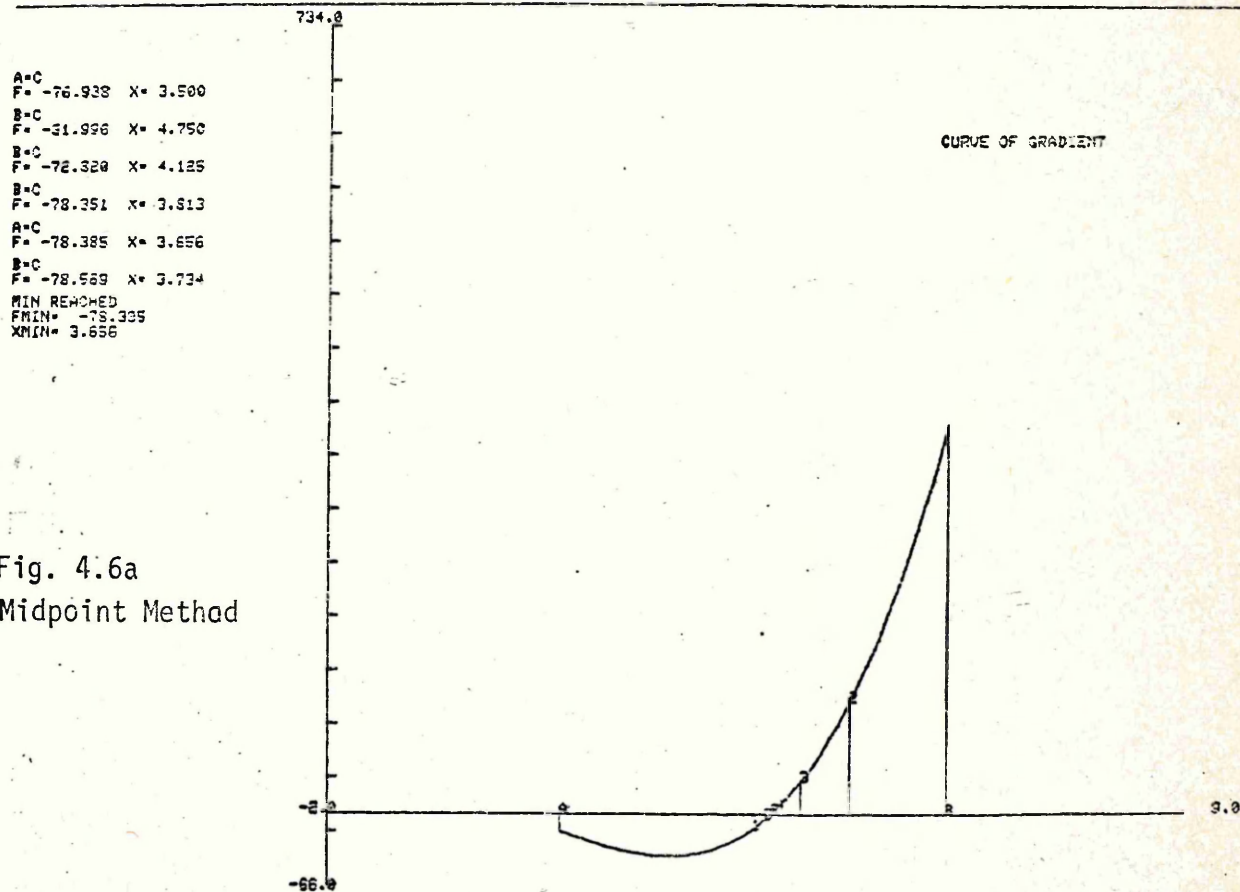


Fig. 4.6a
Midpoint Method



UALS. AFTER 10 EVALS.
F = -77.73 X = 3.572

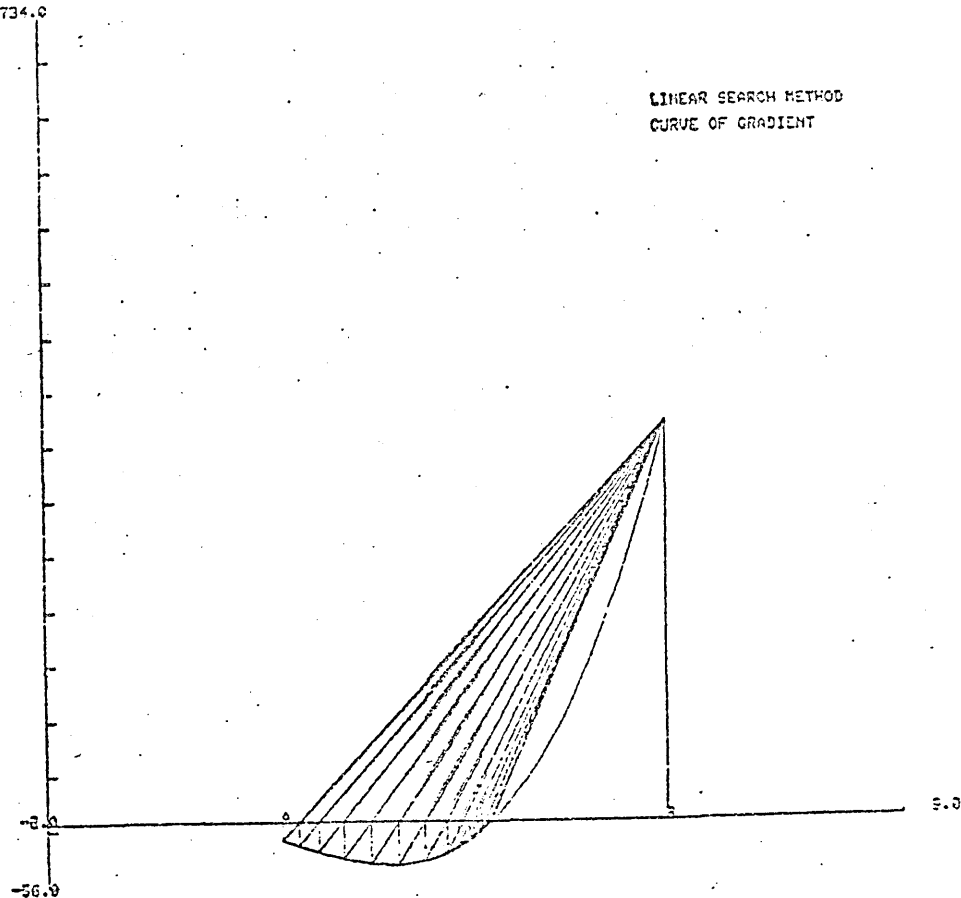
UALS. AFTER 20 EVALS.
F = -78.57 X = 3.731

MIN. REACHED AFTER 23 ITER.
F = -78.57 X = 3.732

LINEAR SEARCH METHOD
CURVE OF GRADIENT

Fig. 4.6b

Linear Search
Method.



PRESS A TO FINISH
PRESS B FOR CURSOR
PRESS C TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

PRESS F FOR LINGO. TCH.
PRESS G FOR CUS. TCH.
PRESS H FOR MIDPT. TCH.

CORD. OF. A = 3.5 -77.1
CORD. PT. B = 3.3 -79.5
CORD. PT. C = 4.0 -75.7

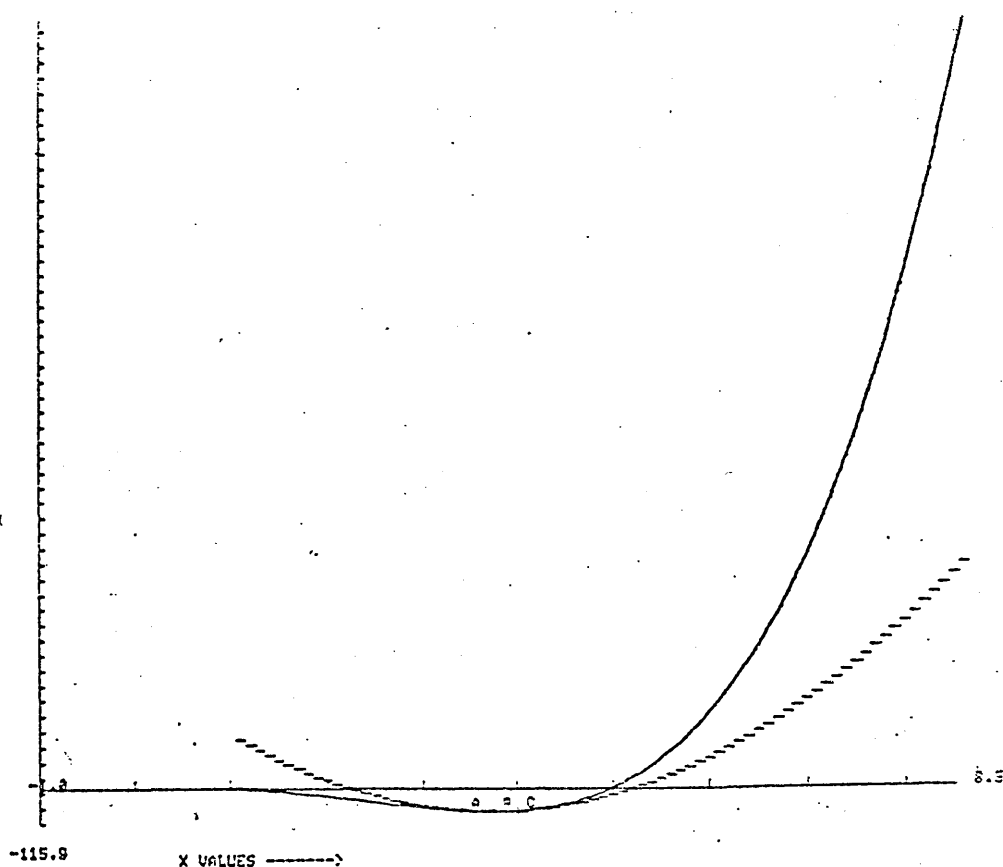
MIN. REACHED
FMIN = -78.491
XMIN = 3.780

2520.4

FUNCTION CURVE

Fig. 4.6c

Davies, Swann &
Campey's
Method.



PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

ROOT OF CUBIC= 3.710
MIN HAS REACHED
FMIN= -78.554
XMIN= 3.710

Fig. 4.6d
Davidon's Cubic
Method.

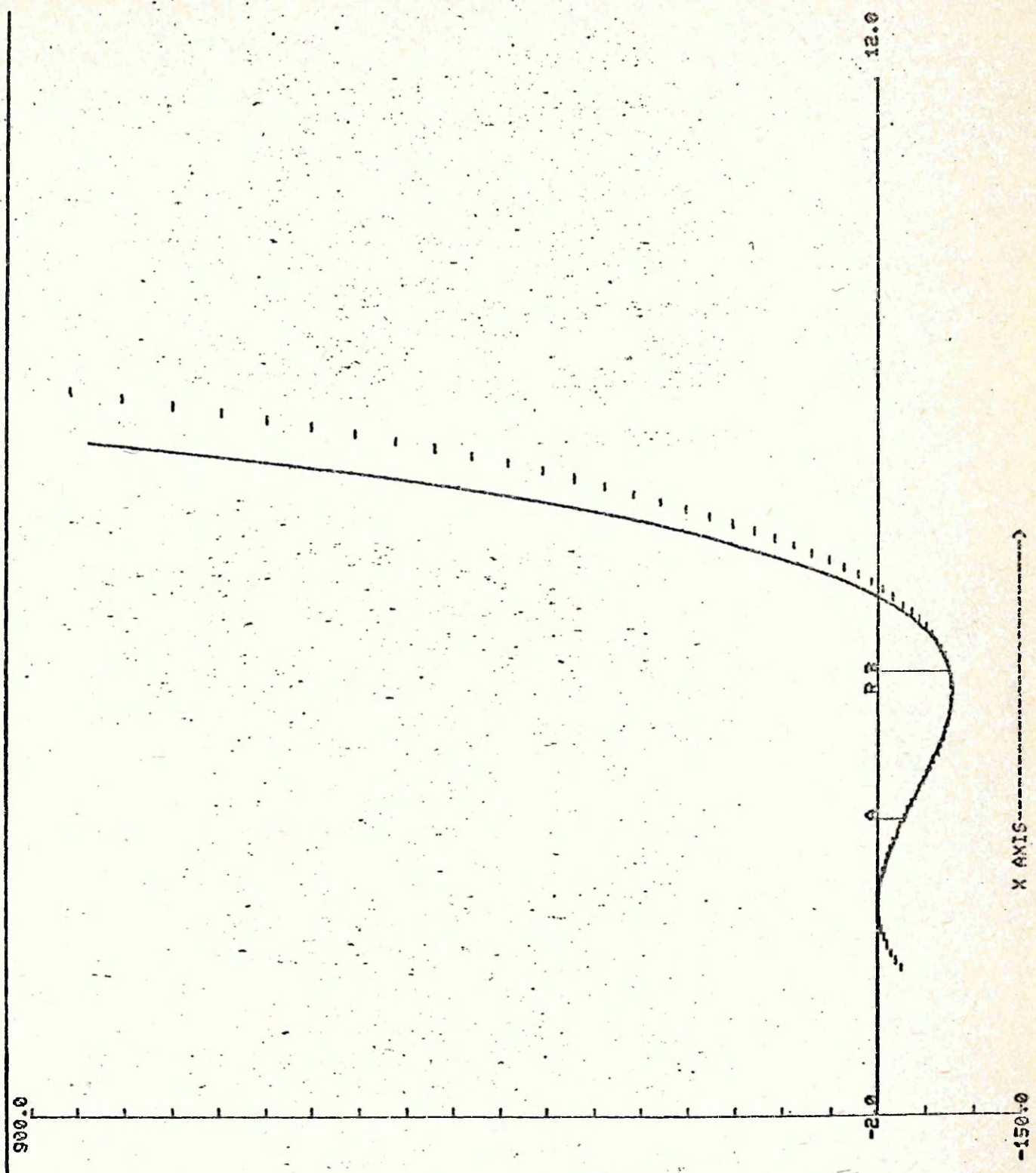


Figure 4.7 shows two contours of the Rosenbrock function (4.1). Figures 4.8(a) and 4.8(b) show contours of Wood's function,

$$F(x) = - \{ 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1(x_2 - 1)^2 + 10.1(x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1) \} \quad (4.3)$$

in the (x_1, x_2) and (x_3, x_4) two-dimensional spaces respectively.

Figure 4.9 presents a constrained function. The dotted closed contour is the non-linear constraint, the straight line with the letter C at both ends is the linear constraint, and the horizontal straight line is the function. The function and the constraints are

$$F(x) = x_2 \quad (4.4)$$

subject to

$$C_1 = x_1^2 + x_2^2 + x_3^2 = 1 \quad (4.5)$$

$$C_2 = 2x_2 - x_1 \leq 1. \quad (4.6)$$

The constraint C_1 is presented in the (x_1, x_2) plane to correspond to the constraint C_2 and the objective function.

A flow chart of the total system is presented in figure 4.10. This shows the decision points and routines available in the system.

In concluding this chapter which has described the principal features of the interactive graphics system, it is useful to list the purposes for which the system was designed and to which it has already been applied, as will be described in the remaining chapters;

- (i) Analysis, evaluation and comparison of one-dimensional search methods, visually and analytically.
- (ii) Analysis, evaluation and comparison of N-dimensional methods on two and N-dimensional problems.
- (iii) Investigation of performance of one-dimensional search methods within N-dimensional methods.
- (iv) Interactively controlling optimization with established methods.
- (v) Investigation of direct interactive optimization of N-dimensional functions, using contour maps only.
- (vi) Presenting contours and investigating optima of constrained problems.

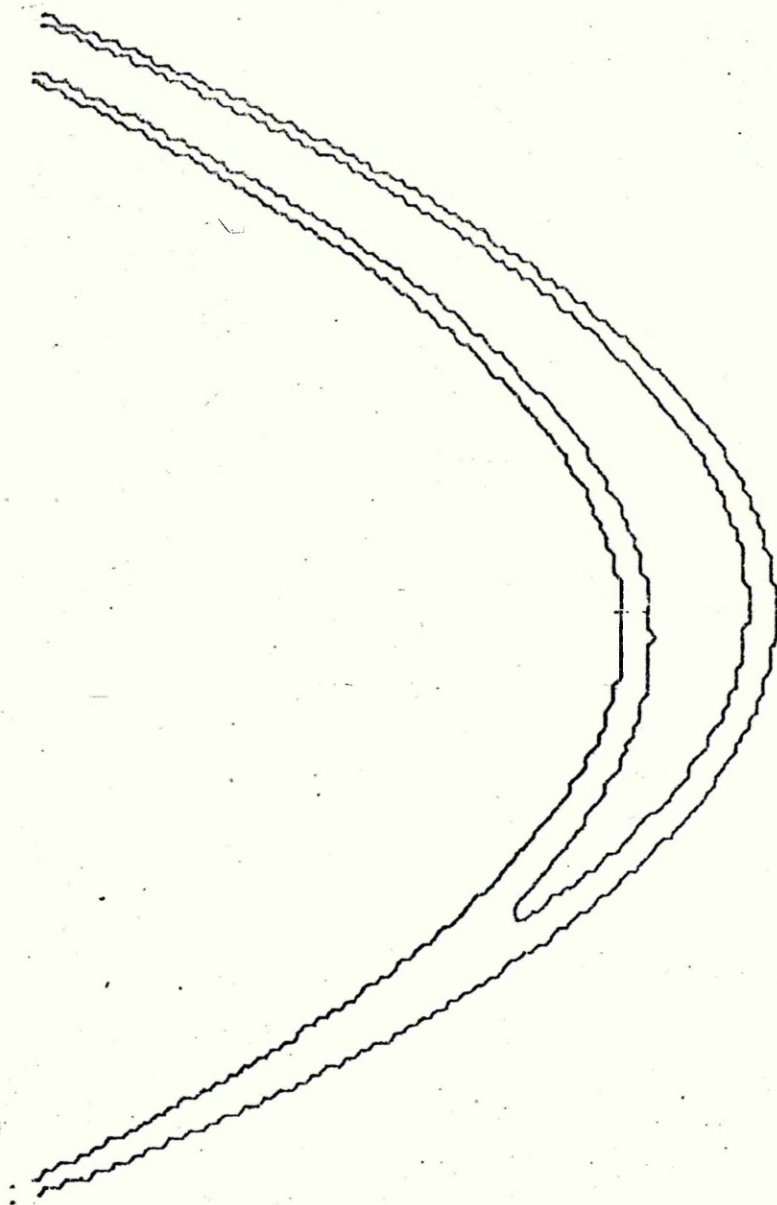


Fig. 4.7
Two Contours of Rosenbrock Function.

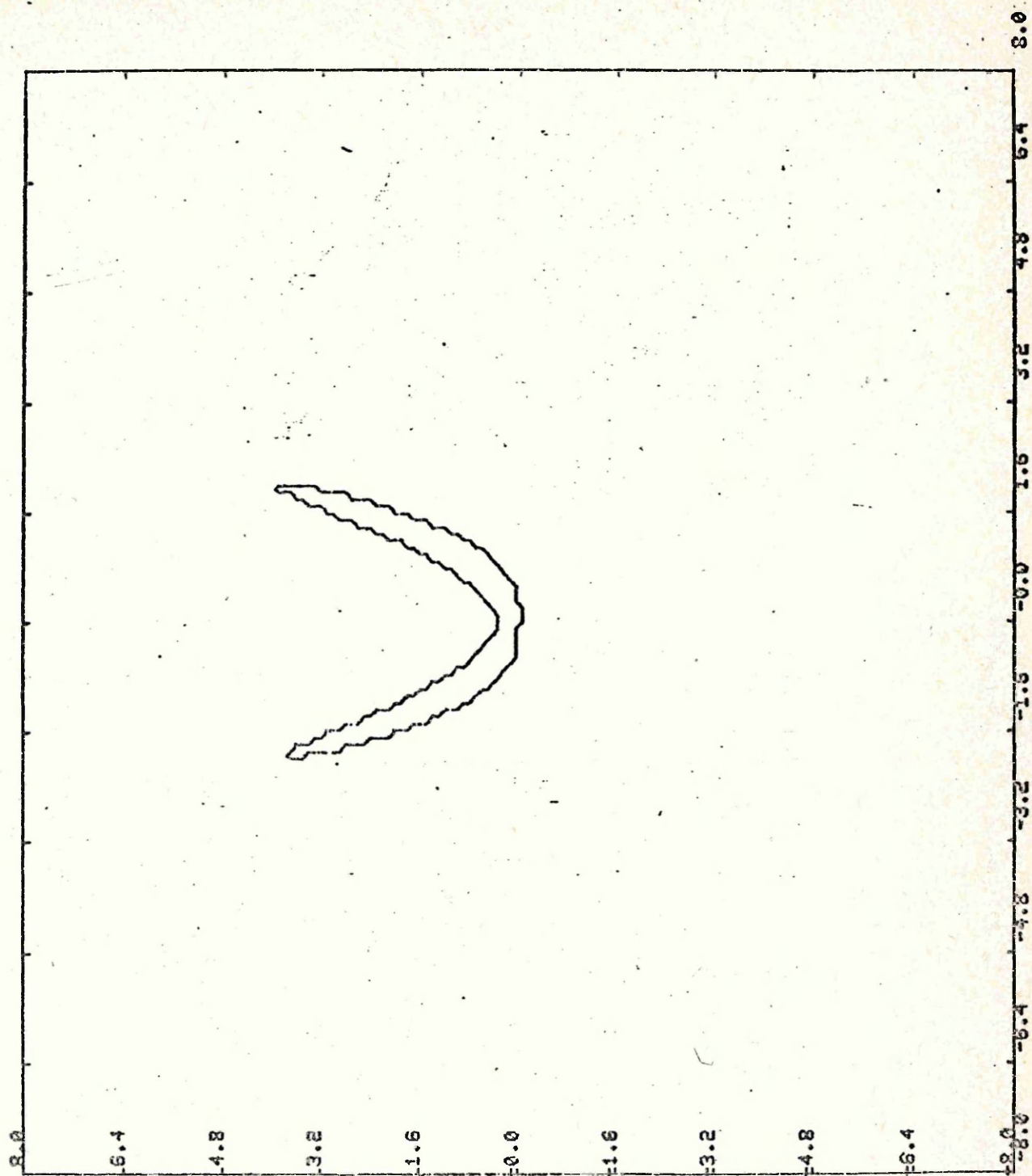


Fig. 4.8a
Contour of Wood's
Function on the
 (x_1, x_2) plane.

FMIN= 42.000
FNC.EVAL. KN1= 568

PRESS M TO CHANGE THE STEP SIZE
PRESS ANY OTHER KEY FOR START

WITH CURSOR, A...J, AS BEFORE

PRESS H FOR NEW PAGE & UNDO
PRESS I FOR NEW PAGE & METHOD
PRESS J STORE MIN. COOR. OF CONTOUR
PRESS K IF ONLY A NEW PAGE IS NEEDED
PRESS L TO CHANGE THE QUADRANT.

PRESS H TO FINISH
PRESS I FOR CURSOR
PRESS J FOR F.P. METHOD
PRESS K FOR MIN. COOR. FINDING
PRESS L, F, G FOR 1, 2, 3RD CONTOUR

THIS PROBLEM DOES NOT HAVE
ANY CONSTRAINTS

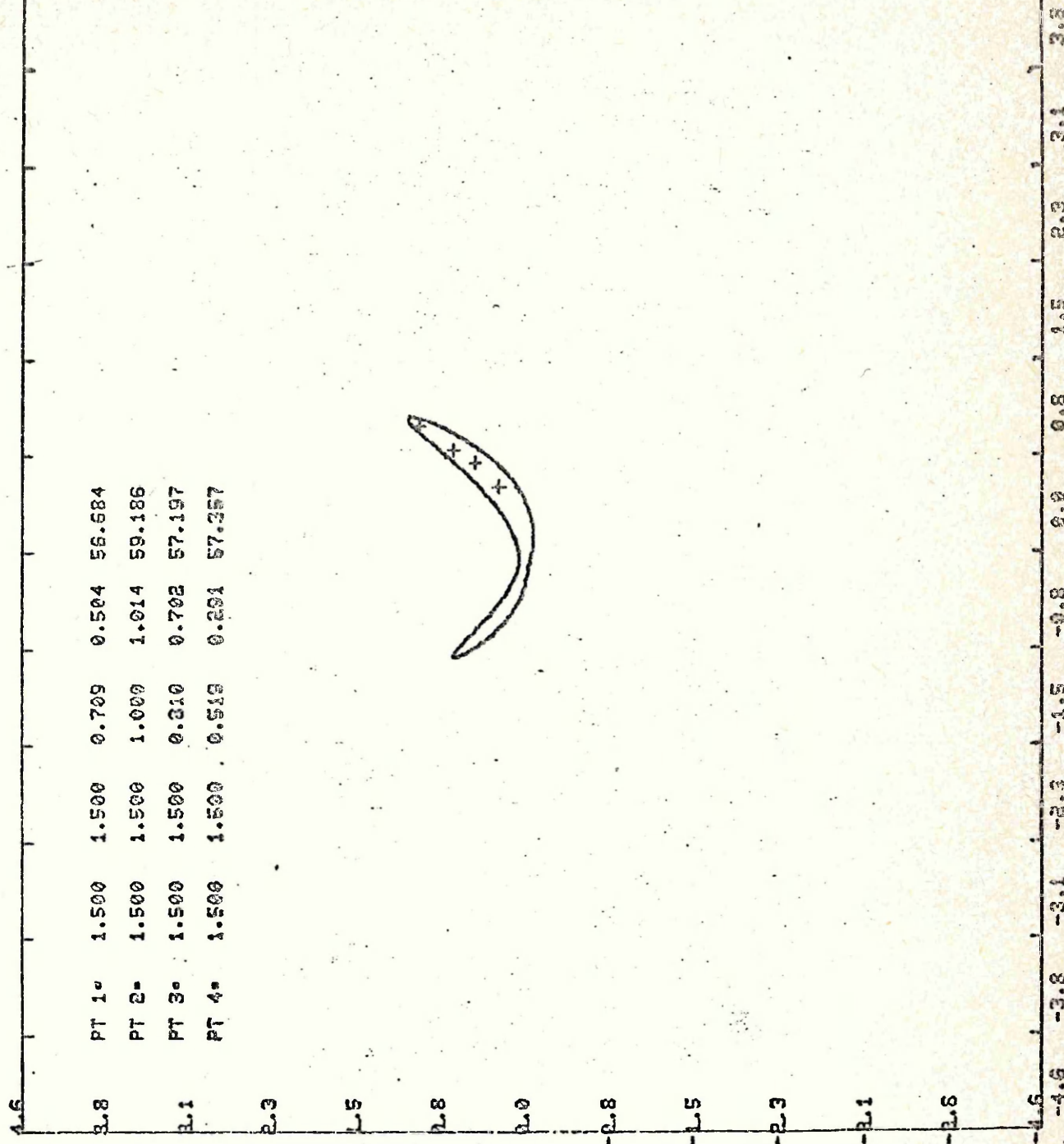


Fig. 4.8b
Contour of Wood's
Function on the
(x_3, x_4) Plane.

FMIN= 60.225
FNC.EVAL. KN1= 1602

PRESS A TO FINISH
PRESS B FOR CURSOR
PRESS C FOR F.P.METHOD
PRESS D FOR MIN COOR. FINDING

PRESS E,F,G FOR 1,2,3RD CONTOUR
PRESS H FOR NEW PAGE & JINDO
PRESS I FOR NEW PAGE & METHOD
PRESS J STORE MIN.COOR. OF CONTOUR

PRESS K IF ONLY A NEW PAGE IS NEEDED
PRESS ANY OTHER KEY FOR START

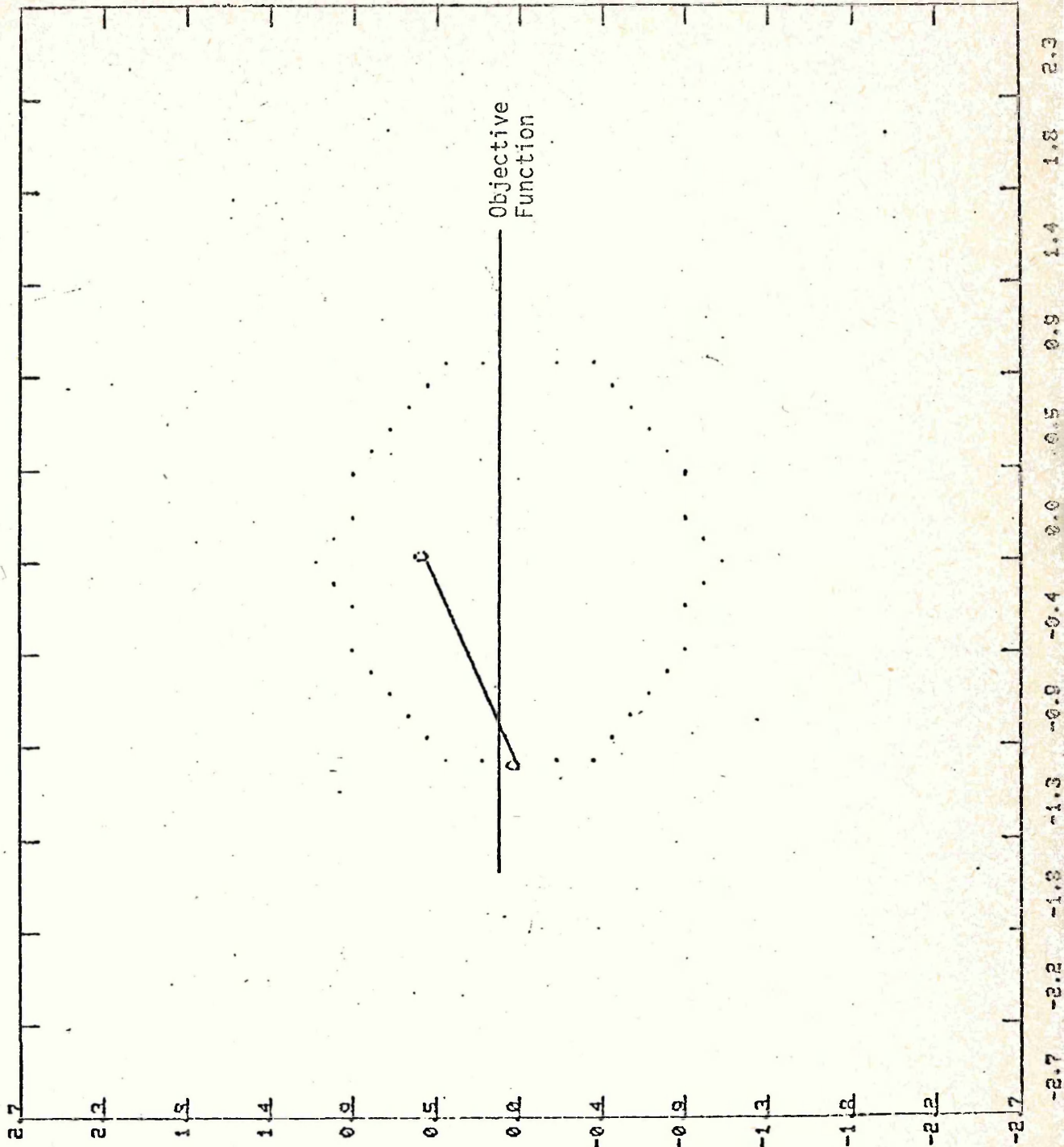


Fig. 4.9
A Constrained
Objective Function.

FNC.EVAL. KNI- 208

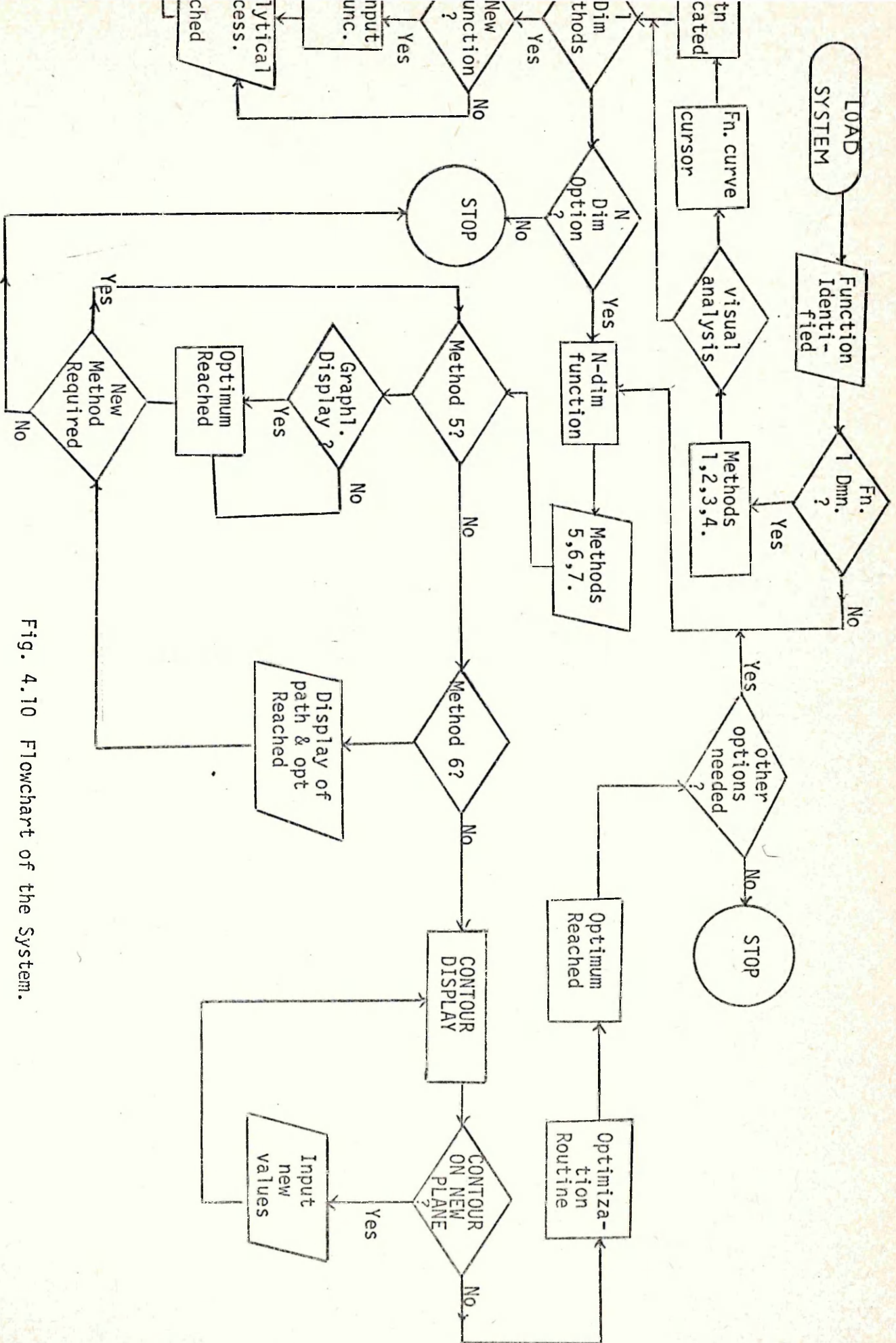


Fig. 4.10 Flowchart of the System.

Having outlined the basic properties and capabilities of the graphics system in the previous chapter, the present chapter goes on to describe in detail some of the practical uses to which it has been put. In each case considered the aim has been to evaluate the contribution which interactive computer graphics might make, either to supplement existing procedures or possibly to replace them.

It was shown in the previous chapter how the system enables a user to select and test procedures for both one and N-dimensional optimization. It was also indicated that facilities are available for the selection of standard or new test functions, as desired.

The first section of this chapter deals with one-dimensional functions and methods. Taking a series of test functions and minimizing them by means of several different procedures, it demonstrates the value, not only of having a flexible interactive system available for this purpose, but also that of being able to produce comparative pictures which are useful in both teaching and research.

Widening the scope to look at problems in more than one dimension, interactive computer graphics can immediately be seen to offer some potential in the case of two-dimensional problems, where contour maps of the function can be drawn and the progress of minimization procedures may be obtained and analysed. Hence, the next section, after describing contouring methods, goes on to illustrate the paths taken by some standard routines on several test functions.

Finally, attention is turned to problems in more than two dimensions, where the contribution of interactive graphics may be less obvious. However, two significant developments are described.

The first consists of inserting an optional graphics feature into an existing N-dimensional algorithm. This enables the univariate search part of the algorithm to be observed and interrupted or modified if desired. The feature is especially useful when an algorithm is causing problems in difficult cases. It offers an insight into the behaviour of the univariate search procedure itself, and also illustrates the joint behaviour of the chosen one and N-dimensional procedures working together. Although the actual running of the program may take longer in c.p.u. time, the user does have a greater degree of control since the system enables the user to terminate a linear search at any time and at any point along the direction of search.

The second area of development consists of investigations into the possibility of using graphics in direct search optimization of functions of more than two variables. Clearly, the image presented can only be in any two of the variables at any time. However, by making systematic slices in the N-dimensional space and observing the resulting contours in two-dimensional displays, some interesting results are obtained. A theoretical investigation of the procedure applied to quadratic forms leads to an algorithm which is, in principle, more generally applicable.

5.1 Problems in One-dimension

There are two commonly used approaches to finding the optimum of a one-dimensional function,

- (i) approximation methods,
- (ii) search methods.

The first approach uses the fact that near that critical point the function may in principle, be expanded using a Taylor series to obtain a better approximation to the optimum by fitting a simple function through a number of computed points. The two methods used here are:-

- (a) Quadratic approximation.
- (b) Cubic approximation.

The search methods are based on finding the roots of $F'(X) = 0$. Many numerical methods follow the same principle, and the two choices for this project are:

- (c) Midpoint method,
- (d) Linear search method.

The general aim is to find the local minimum of a continuous function of X which is differentiable. No further assumptions are made about $F(X)$.

The disadvantages which may occur with either of the two search methods are:-

- (1) it may diverge under undesirable circumstances.
- (2) it may fail to find a local minimum by computational oscillation,
- (3) it may involve an excessive amount of computation for complex functions,
- (4) it depends to a large extent on the initial estimate of the minimum and the search interval.

A graphical presentation will be seen to give a better understanding of the difficulties inherent in the use of these methods, and the optimization pattern followed by each.

5.1.1 A program for one-dimensional problems

This program has been designed to find the minimum of a continuous test function which can be presented in the form of a polynomial, although with minor program modifications more general functions can be used. The system is structured to allow easy incorporation of further, possibly experimental, one-dimensional methods which can then be tested against the ones already included, using a variety of test functions. The system allows the user to choose the method, set up the test function as a polynomial, then input the other requirements such as step size, the initial value of the variable, etc.

The output produced for a method consists of a visual display of either the function curve or the derivative curve and its axes. On the upper right hand corner of the screen the name of the method may be displayed. After an iteration has been performed and visually displayed, the user is given the opportunity to clear the screen and set up the curve and axes given.

When choosing the midpoint or linear search methods the optimum of the test function can be determined in two ways. First, by direct application of the chosen method, in which case after the minimum has been located, the gradient curve is displayed. Alternatively, the function curve may be displayed and the minimum located visually by use of the cursor.

Figure 5.1 shows the flow diagram of the one-dimensional program. When the main program is entered the initial display frame appears automatically. This frame tests the initial instructions and the methods to which the program may branch.

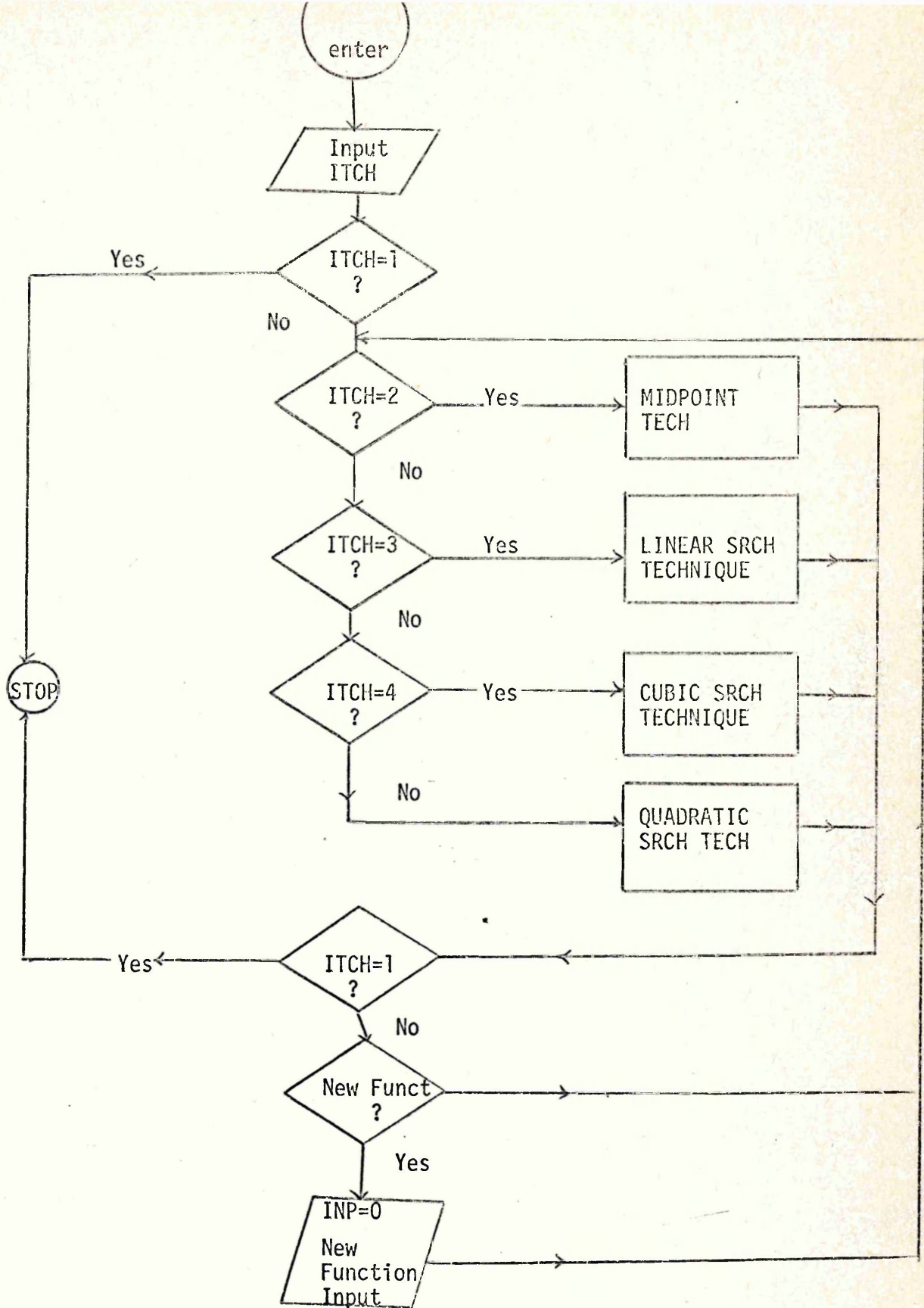


Fig. 5.1 Flow Diagram of the One-Dimensional Program.

When the method is terminated the user is presented with a set of options which is displayed on the left hand side of the screen in the form of a menu, as is shown in figure 5.4.

There are eight options which are commonly used by all four methods. The extra option is only applicable to the linear search and the midpoint methods. This option allows one of the two types of previously mentioned curves to be displayed.

The options can be divided into two categories. The first set operates 'within' the method. For example, the user may view the display more closely by declaring new co-ordinates for the window. It is also possible to test a new function using the method already in use.

The second set of options are those which allow the user to control the overall program by either requesting to use another method, by transferring into the N-dimensional phase, or by terminating the program. Having chosen a new method, the user can test the same function or set up a new one.

Figure 5.2 shows the flow diagram of the methods once the branching off from the main program is done. This diagram also shows the general picture of the layout of the options. Detailed flow diagrams for each method are given later.

(a) Quadratic approximation

This method is based on Davies, Swann and Campey's algorithm and produces three equally spaced points which are used to define a quadratic of the form:-

$$y = ax^2 + bx + c \quad (5.1)$$

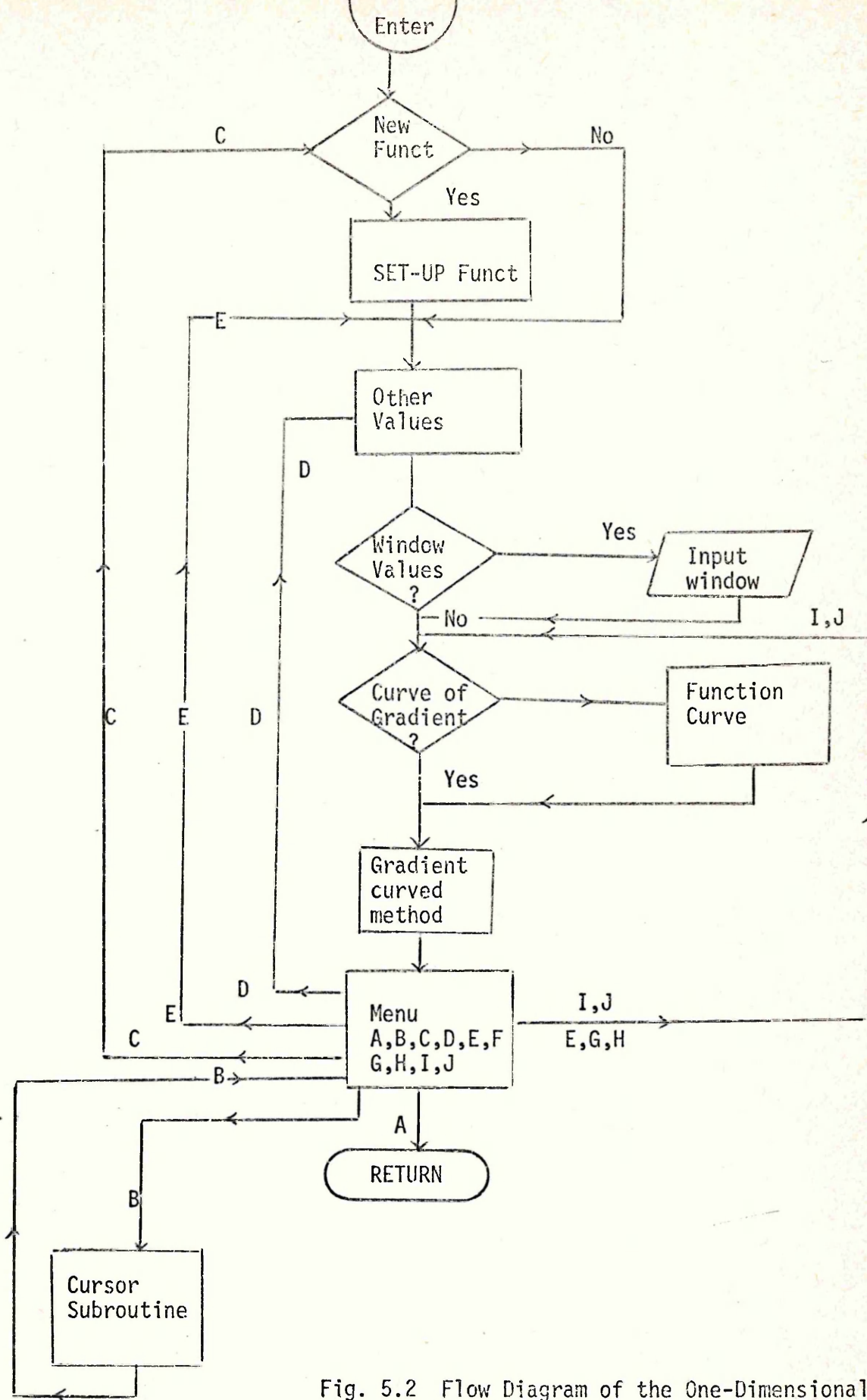


Fig. 5.2 Flow Diagram of the One-Dimensional Methods.

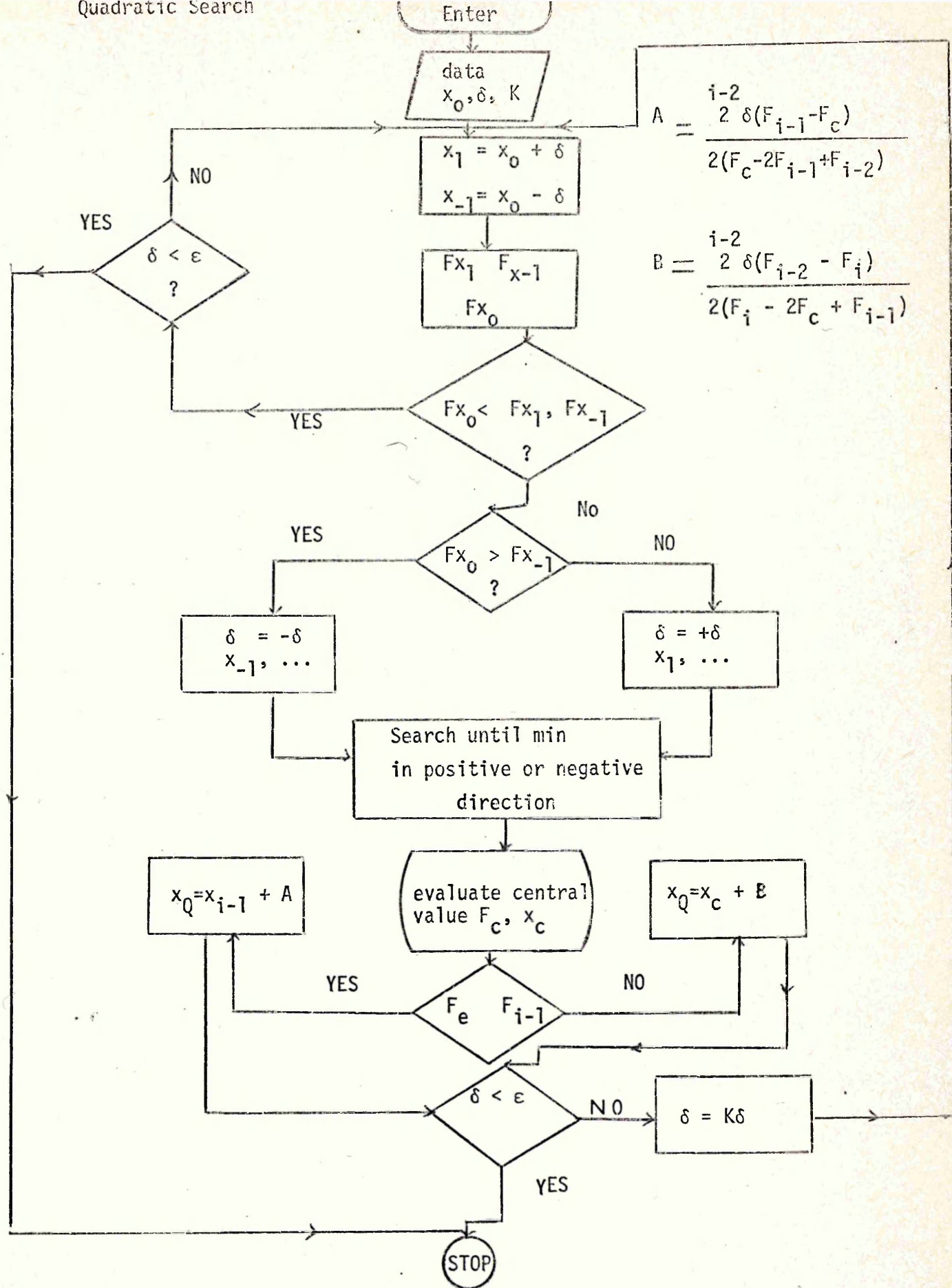


Fig. 5.3 Flow Diagram of Quadratic Method.

Three simultaneous linear equations are obtained by substitution in the above equation, and these are solved to find three unknowns, a, b, and c.

The quadratic approximation to the curve is found using a standard package, SIMQ.

Figure 5.3 shows the flow diagram of the complete method. Figure 5.4 shows the last iteration to find the minimum of the function,

$$F(x) = x^2 - 4x - 6x + 4x + 4. \quad (\text{Function 1})$$

The method is illustrated on the function curve. The three points which span the minimum are indicated by letters A, B and C. These points have been chosen such that $A < B < C$ and $F(A) > F(B) < F(C)$. The co-ordinates of these three points are printed on the left hand side of the screen.

A parabola is fitted through the points A, B and C. This curve is shown as a dashed line and the letter 'R' indicates the predicted minimum. At the end of each iteration the user may produce a hard copy, or proceed, as desired.

This procedure is continued until the magnitude of the distance between two extreme points is sufficiently small, in which case the value of x to give the minimum value of the function is taken as the last value of 'R', that is

$$x_{\min} = R$$

and

$$F(x_{\min}) = F(R).$$

PRESS B FOR CURSOR
 PRESS C TO TEST NEW FUNCT.
 PRESS D TO CHANGE WINDOW
 PRESS E FOR N. INTL. PT.

PRESS H FOR MIDPT. TCH.
 PRESS W FOR VALUE

2520.4

CORD. OF. A= 3.5 -77.1
 CORD. PT. R= 3.8 -78.5
 CORD. PT. C= 4.0 -75.7

MIN. REACHED
 FMIN= -78.491
 XMIN= 3.780

FUNCTION VALUE

Fig. 5.4
 Quadratic
 Approximation
 Method.

0.5

1.0

A B C

-115.9

X VALUES

(b) Davidon's cubic technique

This method obtains the minimum value of a function by approximating it to a cubic over its critical part, using both the value and the derivative of the function at two distinct points.

The value of the initial point A and the step size H have to be selected at the start of the procedure. The next step is to calculate the value of the derivative $F'(A)$ at point A. If $F'(A) < 0$ then, $H_1 = -H$ otherwise, $H_1 = H$. A new point B is found and its derivative $F'(B)$ is calculated, where $B = A + H_1$. The sign of $F'(B)$ is compared to that of $F'(A)$. If they have equal signs, the point B replaces A as the new origin, the step size H_1 is doubled and the above process is repeated to find a new origin. If the signs differ, or the value of the derivative at point B is zero, then a cubic is fitted to the points A and B using the previously calculated values of $F(A)$, $F'(A)$, $F(B)$ and $F'(B)$. The cubic is of the form;

$$y = ax^3 + bx^2 + cx + d \quad (5.4)$$

where a, b, c and d are unknown. They may be found by solving a set of four simultaneous equations, where for convenience the origin is taken at point A. The four equations are;

$$\begin{aligned} F(A) &= d \\ F'(A) &= c \\ F(B) &= ax^3 + bx^2 + cx + d \\ F'(B) &= 3ax^2 + 2xb + c \end{aligned} \quad (5.5)$$

where $x = H$.

Having solved these four linear equations using SIMQ, the minimum value of this cubic can be estimated. This minimum value is then taken as an approximation to the minimum value of the test function. This point, referred to as 'R' replaces either point A or the point whose derivative has the same sign as $F'(R)$. The above procedure is repeated until the absolute value of the derivative of the root of the cubic approximation $F'(R)$ is less than a prescribed value. Figure 5.5 shows the flow diagram of this method.

Figure 5.6 presents a graphical illustration of this method used to obtain a minimum for function (1).

It shows the function and the two points A and B after a cubic fit, and the letters A and B specify the position of the two points on the x axis. Two vertical lines are also drawn from $F(A)$ and $F(B)$ to the x axis. These two vertical lines show the values of $F(A)$ and $F(B)$ on the graph.

A cubic is fitted at the points A and B. This curve is drawn using a dashed line. The minimum of this curve is labelled 'R' and a line is drawn from this point on the x axis to the curve which indicates the position of $F(R)$. This minimum value is taken as an approximation of the minimum of the function. The value of minimum point 'R' and its function value are displayed.

If more than one cubic fit is required before the minimum is reached, the process is stopped after each iteration to allow the user to observe the output or to produce a hard copy before continuing the process.

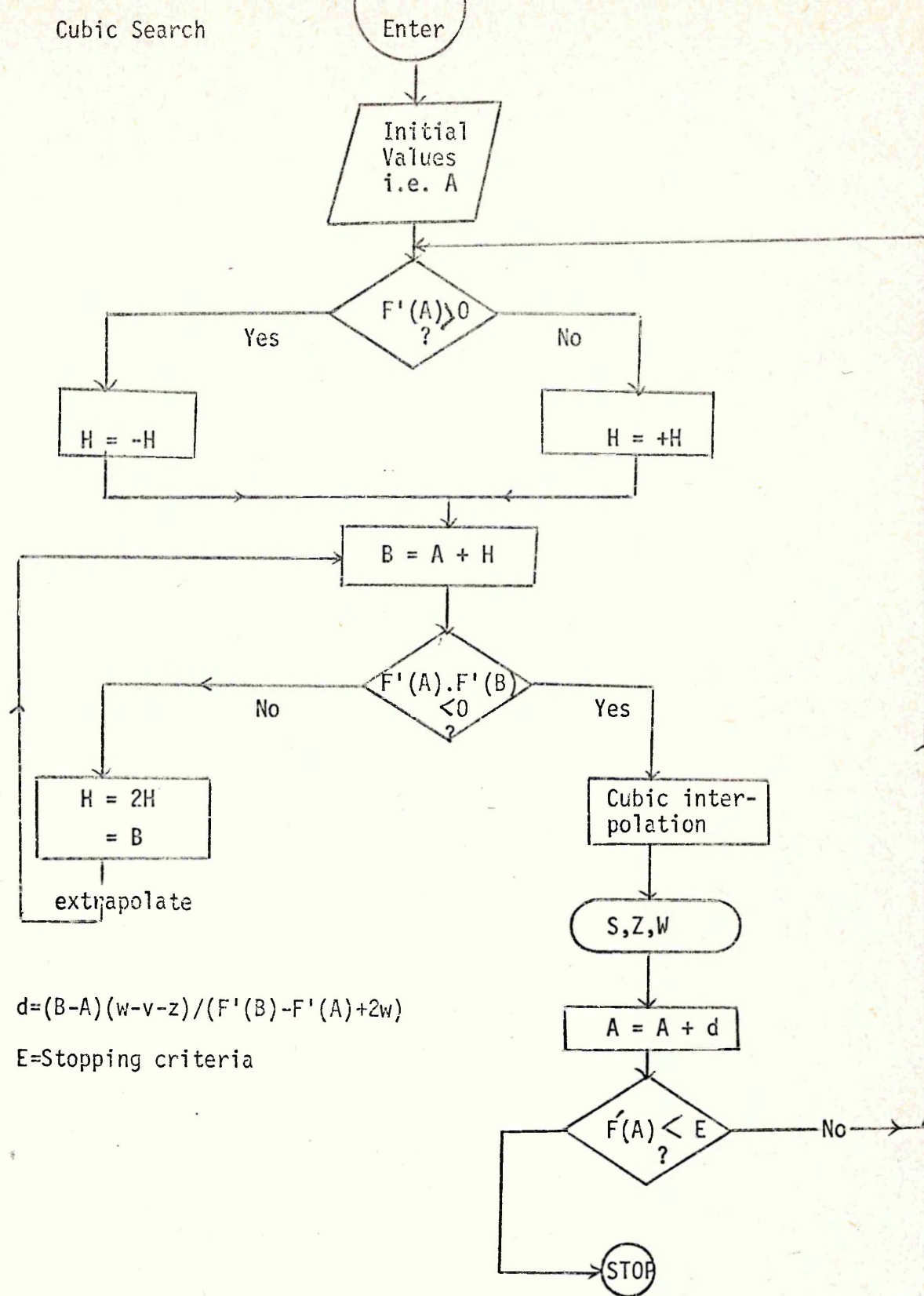
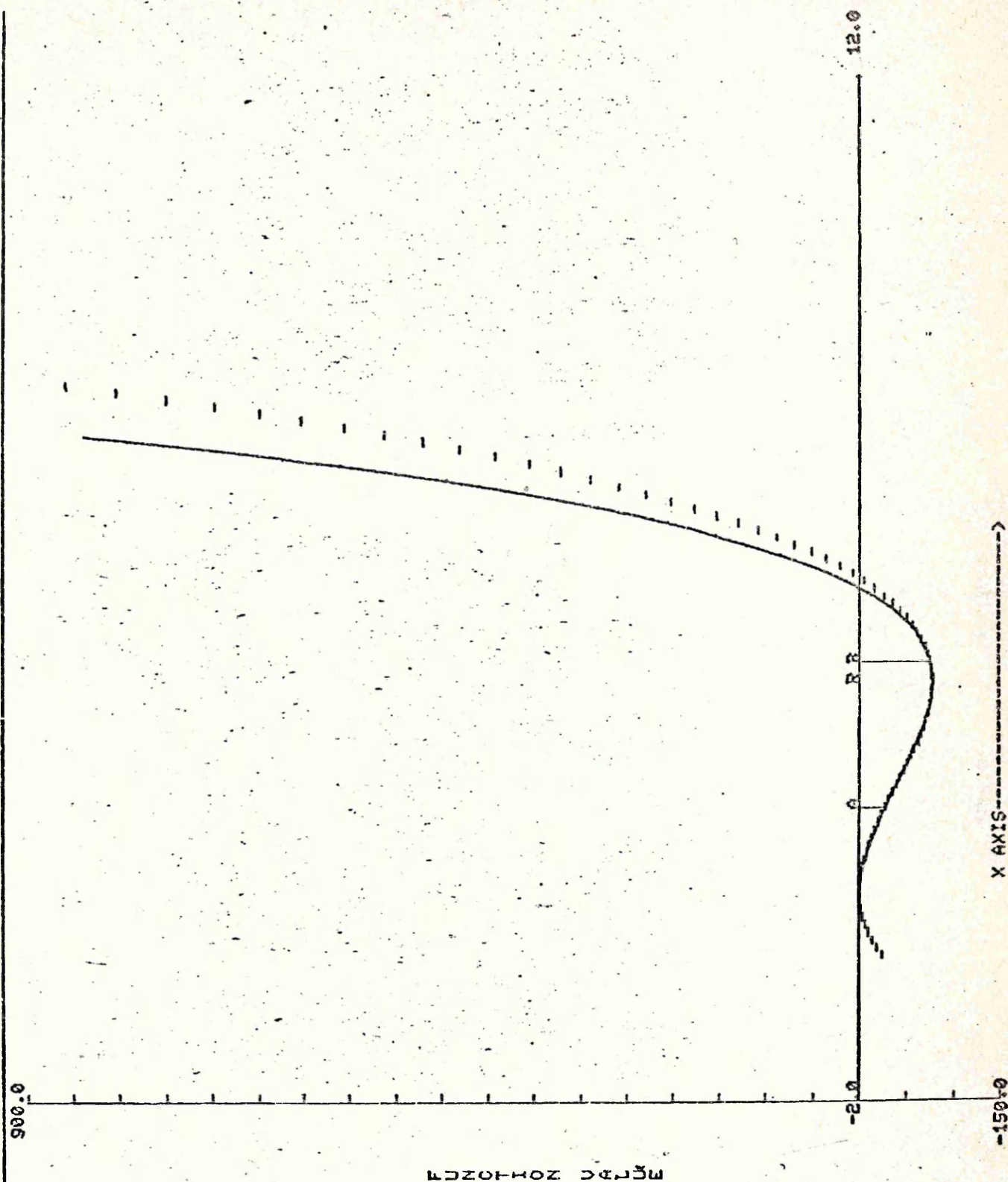


Fig. 5.5 Flow Diagram of Cubic Search Method.

PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

ROOT OF CUBIC= 3.710
MIN HAS REACHED
FMIN= -78.554
XMIN= 3.710

Fig. 5.6
Display of Cubic
Search Method.



Once the minimum is reached the minimum point and its function value are displayed on the left hand side of the screen.

(c) The midpoint search technique

This method is also known as the "Method of bisection". It requires two points A and B to specify the search interval of variable x. These two points should be chosen so that $A < B$. The values of the derivative of the function at these two points $F'(A)$ and $F'(B)$ are calculated and a check is made to ensure that they are of opposite signs. If this condition is satisfied then $F'(x)$ must have a root or a discontinuity within this search interval.

The value of the derivative is calculated for a third point C, where $C = (A+B)/2$, $F'(C)$ is then compared with $F'(A)$ and $F'(B)$. If $F'(C)$ has the same sign as $F'(A)$, the point C replaces point A, otherwise the point C becomes the new point B. This process is repeated until the magnitude of the distance between the current points of A and B or the value of $F'(C)$ is sufficiently small, in which case the value of the root of $F'(x)$ will be the last value of C, that is

$$x_{\min} = C, \quad \text{and} \quad F(x_{\min}) = F(C).$$

Figure 5.7 shows an illustration of this method to find the minimum value of function (1).

The curve is that of the derivative of the function. On the curve, the search interval specified by the points A and B is indicated by drawing two vertical lines, one from $F'(A)$ and the other from $F'(B)$ to the x axis. The actual points on the

curve are shown by the letters A and B on the x axis at the position of the points A and B respectively. These two vertical lines are not numbered, indicating that no iteration has yet been performed. The next line drawn represents the midpoint of the interval AB at the point C. A vertical line is drawn from $F'(C)$ to the x axis. This line is given the number '1', which signifies the result of the first iteration. On the left hand side of the screen the value of C which is assigned to x, the function value and the point which has been replaced by C are printed. Each new midpoint which is drawn using the above method is annotated with its iteration number. The numbers on the curve also give an indication of how the interval has been reduced.

This illustration shows that six iterations have been performed before a minimum has been reached.

(d) The linear search technique

This method starts by initially finding two points, A and B such that $A < B$ and $F'(A)$ and $F'(B)$ are of opposite signs. It is assumed that the derivative curve in the neighbourhood of its root can be approximated by a straight line, whose slope and intercept are linear combinations of $F'(A)$ and $F'(B)$. The point where this line crosses the x axis is taken as an approximation to the root of the derivative curve. With respect to this assumption a point C is found where C is given by the expression

$$C = A - ((B-A)F'(A)/(F'(B)-F'(A))) \quad (5.6)$$

The value of $F'(C)$ at this point is calculated and its sign is compared with that of $F'(A)$ and $F'(B)$. If $F'(C)$ has the same

sign as $F'(A)$ the point C becomes the new point A, otherwise point C will replace point B. The process of obtaining a third point is repeated until the magnitude of the distance between the two current points A and B has become smaller than a predetermined tolerance, in which case the value accepted as the root of $F'(x)$ is the value of C.

Both the linear search and midpoint methods are based on the same basic principle, except for the way in which they obtain the third point C. Figure 5.8 shows the flow diagram for both methods.

A graphical illustration of the linear search method for function (1) is shown in figure 5.9a. As for the previous method the search interval is specified by drawing two vertical lines from $F'(A)$ and $F'(B)$ to the points A and B on the x axis. A straight line joins the current $F'(A)$ at the point A to the current $F'(B)$ at point B, after each iteration until a value for the root is obtained or the method is interactively terminated by the user on the basis of displayed evidence.

At the start of the method the user is requested to specify the number of iterations after which the current values of the function, the gradient and the variable are to be printed out on the left hand side of the screen for inspection. At this stage, the user is given the choice of continuing the process, or terminating it if the results are not satisfactory.

If the starting condition, that is $F'(A)$ and $F'(B)$ should have the opposite sign, is not initially satisfied, then the user is requested to submit a new range. This process can continue until the starting condition is met.

A=C -76.938 X= 3.500
 B=C -31.996 X= 4.750
 B=C -72.820 X= 4.125
 B=C -78.351 X= 3.813
 A=C -78.335 X= 3.656
 B=C -78.569 X= 3.734
 MIN REACHED
 FMIN= -78.385
 XMIN= 3.656

734.0

CURVE OF GRADIENT

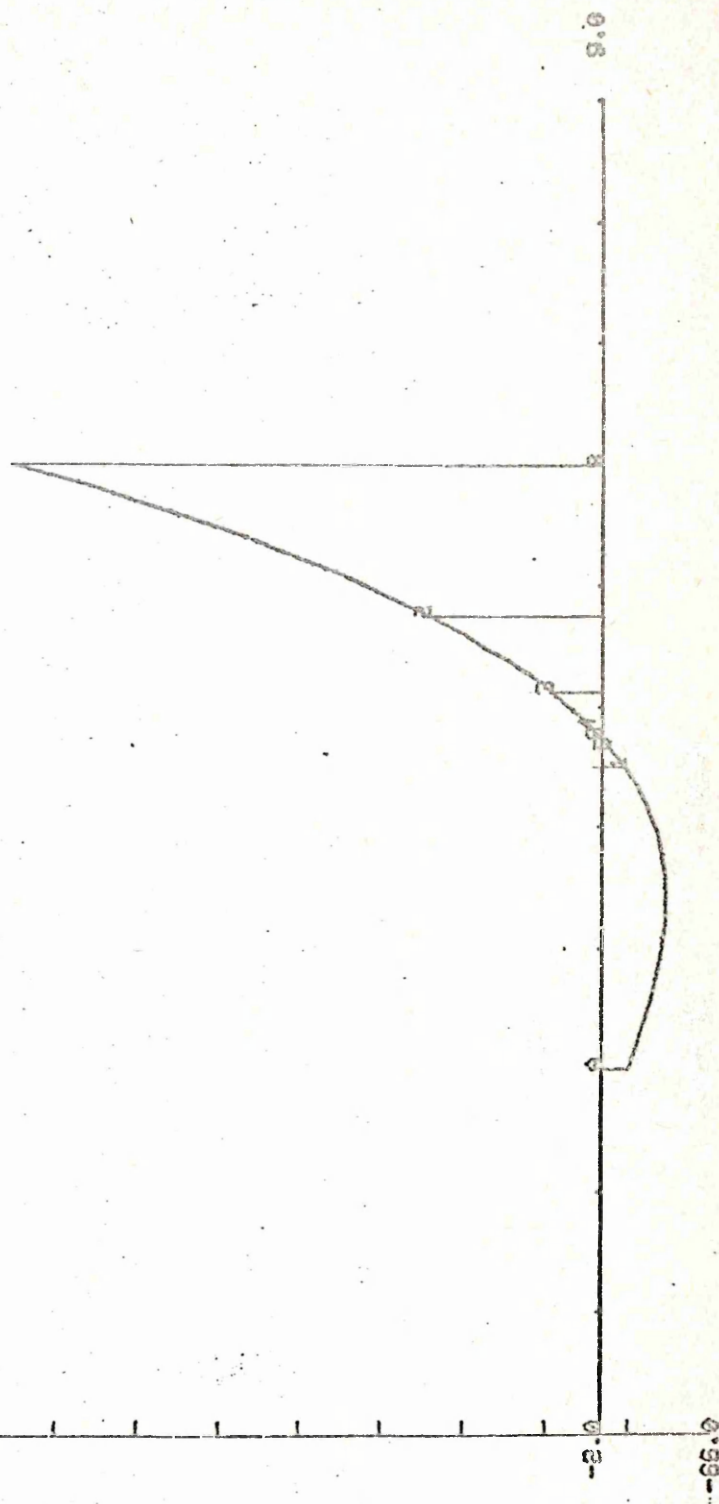


Fig. 5.7

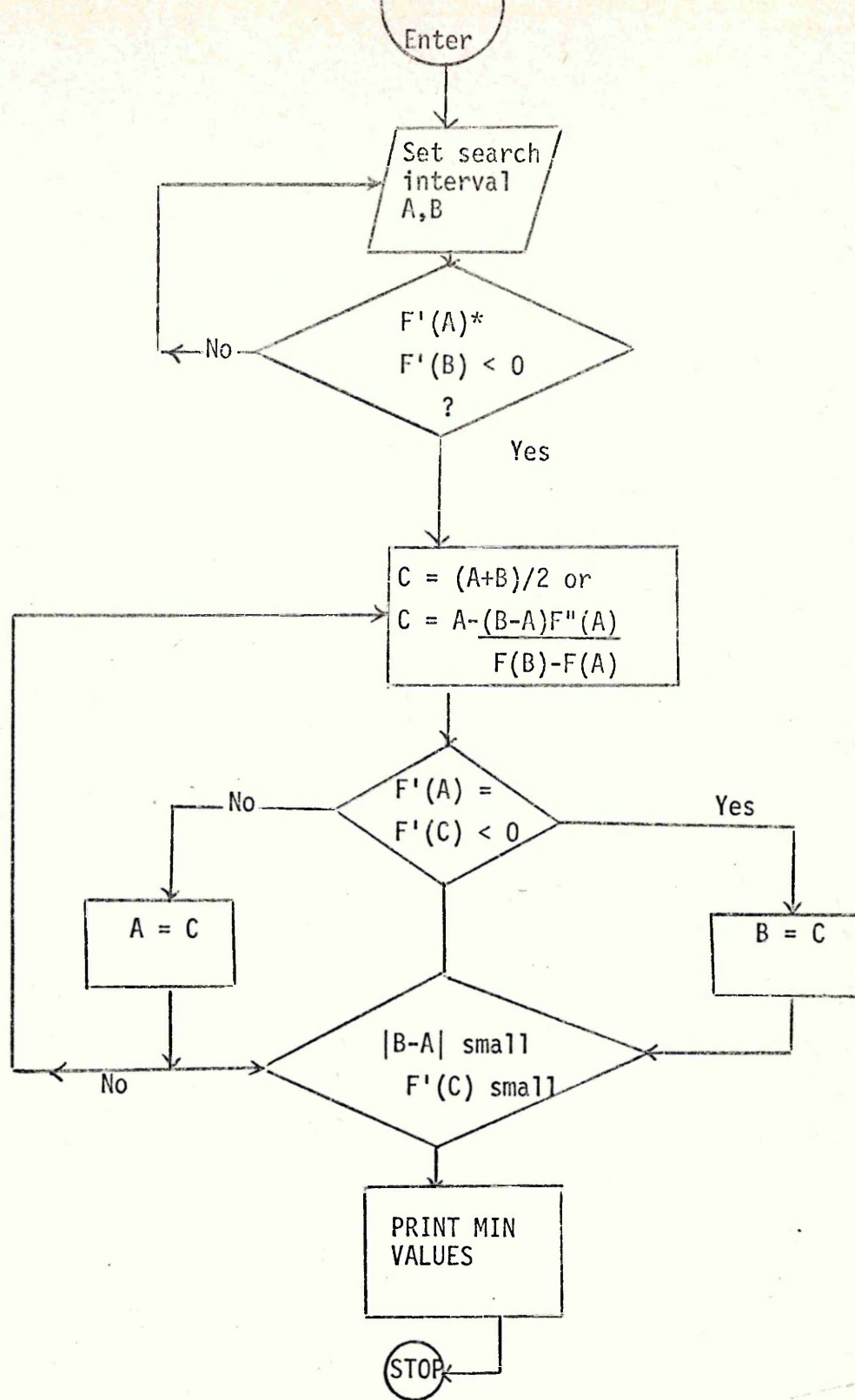


Fig. 5.8 Flow Diagram of Linear Search and Mid-Point Techniques.

PRESS B FOR CURVE
PRESS C TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

PRESS G FOR LUB.
PRESS H FOR QUAD. TCH.
PRESS I FOR FUCTION CURVE

UALS. AFTER 10 EVALS.
F = -77.78 X = 3.572

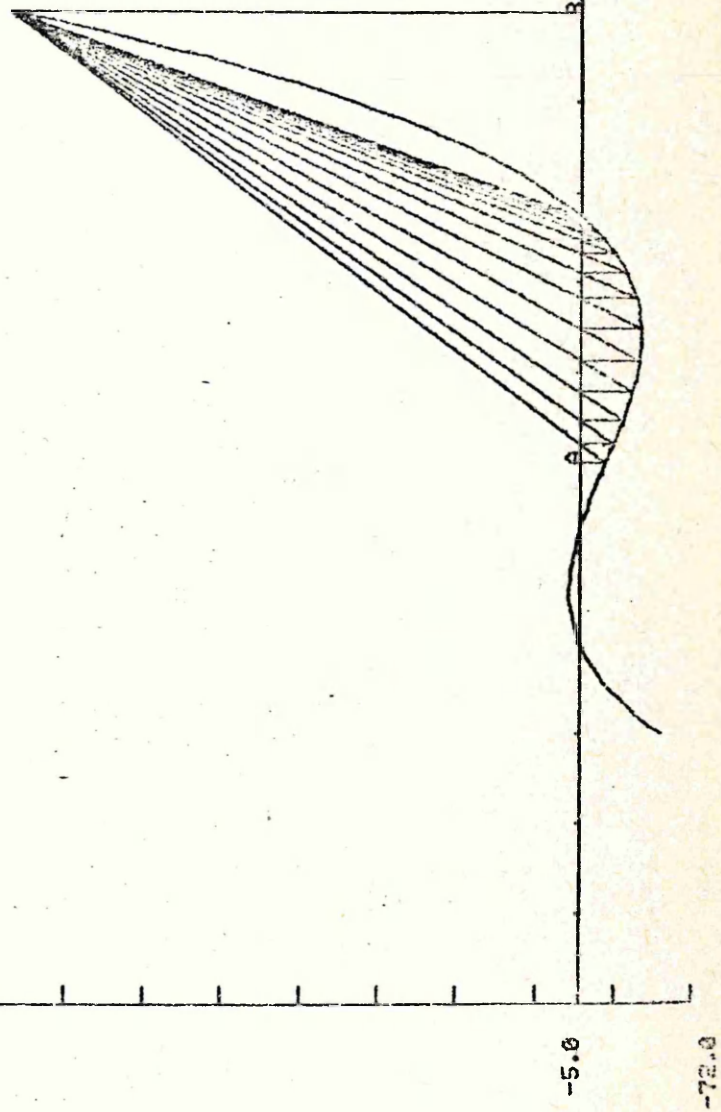
UALS. AFTER 20 EVALS.
F = -78.57 X = 3.731

MIN. REACHED AFTER 23 ITER.
F = -78.57 X = 3.732

734.0

LINEAR SEARCH METHOD
CURVE OF GRADIENT

Fig. 5.9a
Linear Search
Technique.



PRESS E FOR CURSOR
PRESS F TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

PRESS G FOR J1, J2, J3
PRESS H FOR QUAD. TECH.
PRESS J FOR GRADIENT CURVE

797.0

CURVE OF FUNCTION

FUNCT VAL. = -78.57 AT 3.742
FUNCT VAL. = -77.92 AT 3.870

-5.0

-105.0

Fig. 5.9b

Min. Value of Function
Located Visually.

The minimum value for the function (1) is reached after twenty three iterations, which indicates that the method is slow in obtaining the minimum when compared to the previous method, where the minimum was reached after six iterations.

5.1.2 Test functions

Five test functions have been chosen to test the four one-dimensional methods available. Two of the functions are the ones referred to by Dixon (23).

Function 1

$$F(x) = x^4 - 4x^3 - 6x^2 + 4x + 4 \quad (5.7)$$

The illustrations of the four methods to find the minimum of this function have already been presented. The true value of the root of the derivative of this function is 3.73205. If this value is compared to the results produced by the four methods, it can be seen that the minimum obtained by the linear search method, $x = 3.732$, shown in figures 5.9(a), is the nearest to the true value. However, an approximate minimum can also be obtained analytically as shown in figure 5.9(b), where the user predicts the minimum using the cursor. The value obtained is acceptable when compared with the other results produced by optimization methods.

Function 2

$$F(x) = -3x^4 + 44x^3 - 120x^2 - 384x + 1$$

This function has a minimum at 4.0. Figures 5.10(a), (b) show the minimum found by the mid-point, linear search, Figures 5.10(c) and 5.10(d), Davies, Swann and Campey methods respectively.

For this particular function within the search region of 1 and 6, the linear search method has proved to be more suitable than the others. This method obtains the minimum in one iteration with the exact value of $x = 4.0$.

Function 3

$$F(x) = x^2$$

This function has a minimum at the point $x = 0.0$, and all four methods are illustrated on this function. Figures 5.11 (b), (c), (d) and (e) show the graphical illustrations of the midpoint, linear search, Davies, Swann and Campey and Davidon's cubic methods respectively. The minimum as predicted by the linear search method is again more accurate and only one iteration is used. The midpoint method however, required seven iterations before the minimum was found and this value is not as accurate as the value found by the other three methods. The performance and the minimum found by Davies, Swann and Campey and Davidon's cubic method are relatively similar.

Function 4

$$F(x) = x^4 - 4x^3 - 6x^2 - 16x + 4$$

The minimum of this function occurs at the point $x = 4.0$. The other roots are imaginary. Figures 5.12 (b), (c), (d) show the three methods, midpoint, linear search and Davies, Swann and Campey, used to find the minimum value of this function. Figure 5.12(a) shows the minimum predicted visually. This value is reasonably accurate when compared with the minimum found by numerical methods.

For this function Davidon's method, figure 5.12e finds the minimum in one cubic interpolation to a reasonable degree of accuracy.

Function 5

$$F(x) = x^4 - 4x^3 - 6x^2 + 64x + 4.$$

The minimum of this function is at -2.0. Figures 5.13 (a), (b), (c) and (d) show graphical illustrations of the midpoint, linear search, Davies Swan and Campey's and Davidon's method.

From the five test functions and the results obtained by each method, it is apparent that for each function a different method was found to be more suitable, depending on the starting values. However, the performance of the midpoint technique was found to be consistent for all five functions. In all five cases, it required seven or eight iterations to locate the approximate minimum and the minimum values were generally less accurate than those found by other methods.

5.2 Problems in Two-dimensions

The problem is to find the values of x_1 , x_2 which give the least value of $f(x_1, x_2)$.

It is possible to visualise a two dimensional function in terms of a landscape where f is the height of land at the point (x_1, x_2) and to plot contours at equal increment of f . The problem of minimization is to locate the lowest point on the surface of the landscape.

There are many optimization methods which are used to find the minimum of a two-dimensional function, such as Fletcher and Reeves

$X = 4.500$
 $A-C$
 $X = 3.750$
 $A-C$
 $X = 3.625$
 $B-C$
 $X = 4.063$
 $A-C$
 $X = 3.244$
 $A-C$
 $X = 3.953$
 $B-C$
 $X = 4.000$
 MIN. REACHED
 $FMIN = -1.26.735$
 $XMIN = 3.653$

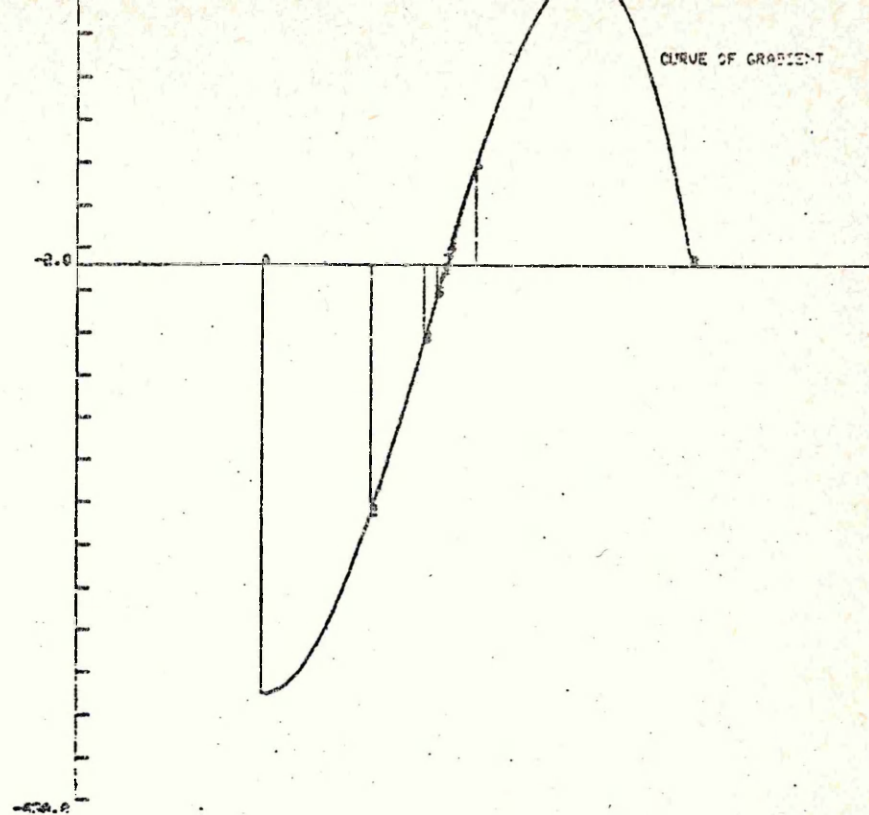


Fig. 5.10a
 Function (2)
 Midpoint
 Technique.

PRESS A TO FINISH
 PRESS F FOR CURSOR
 PRESS C TO TEST NEW POINT.
 PRESS D TO CHANGE WINDOW
 PRESS E FOR N. INT. PT.

PRESS F FOR RED. TCH.
 PRESS G FOR CUB. TCH.
 PRESS H FOR QUAD. TCH.
 PRESS I TO FINEST CURVE

MIN. REACHED AFTER 1 ITER.
 $F = -1497.29$ $X = 4.000$

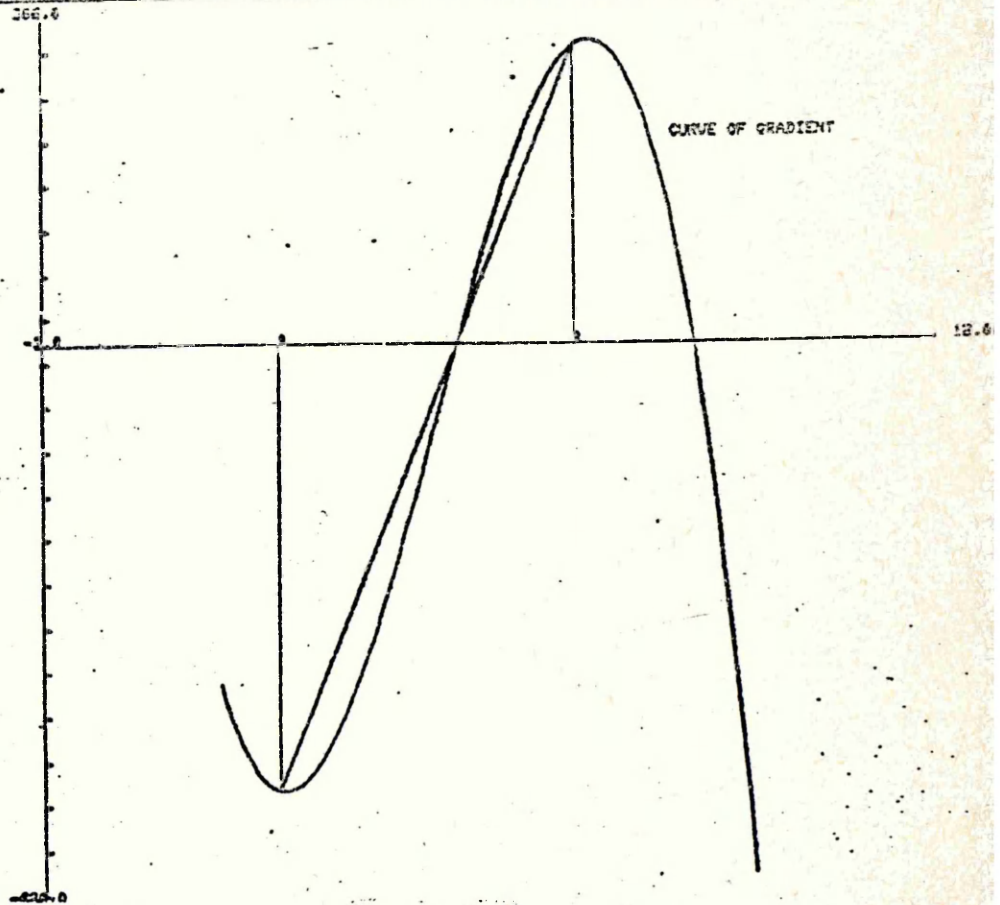
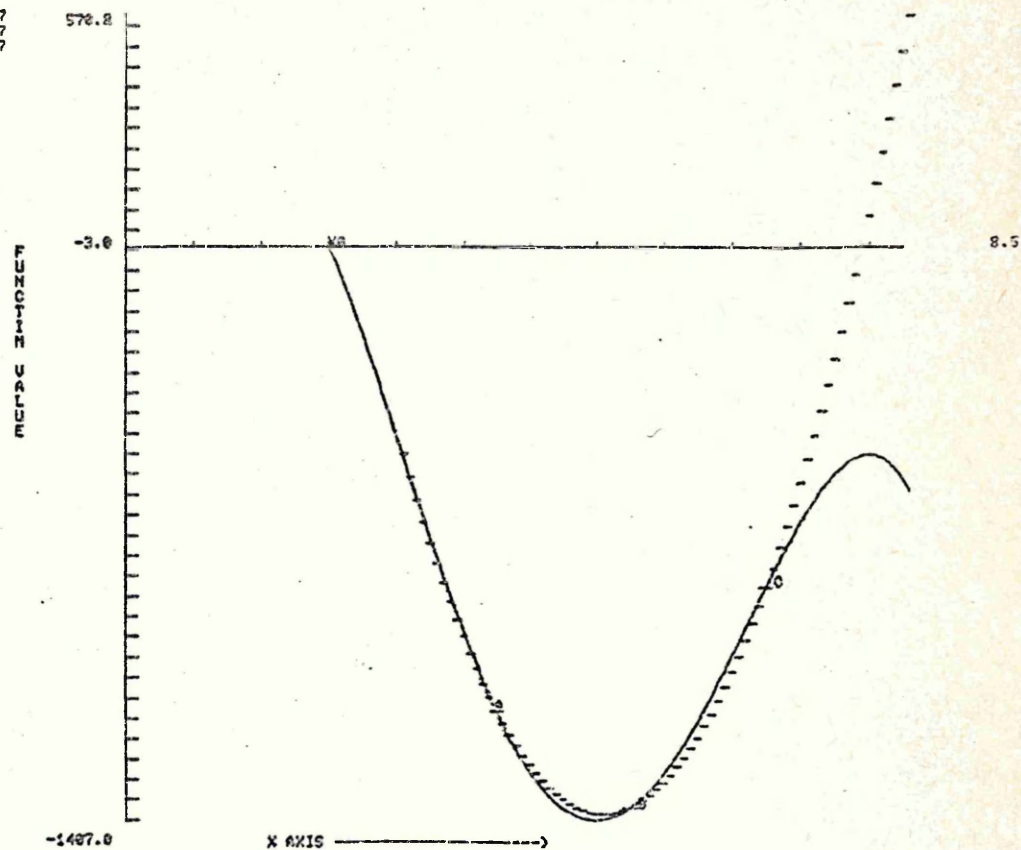


Fig. 5.10b
 Linear Search
 Method.

CORD. OF. A= 2.5-1138.7
 CORD. PT. B= 4.5-1377.7
 CORD. PT. C= 6.5-836.7
 PRESS C TO CONTINUE



CORD. OF. A= 2.0-943.0
 CORD. PT. B= 4.0-1467.0
 CORD. PT. C= 5.0-1007.0
 PRESS C TO CONTINUE

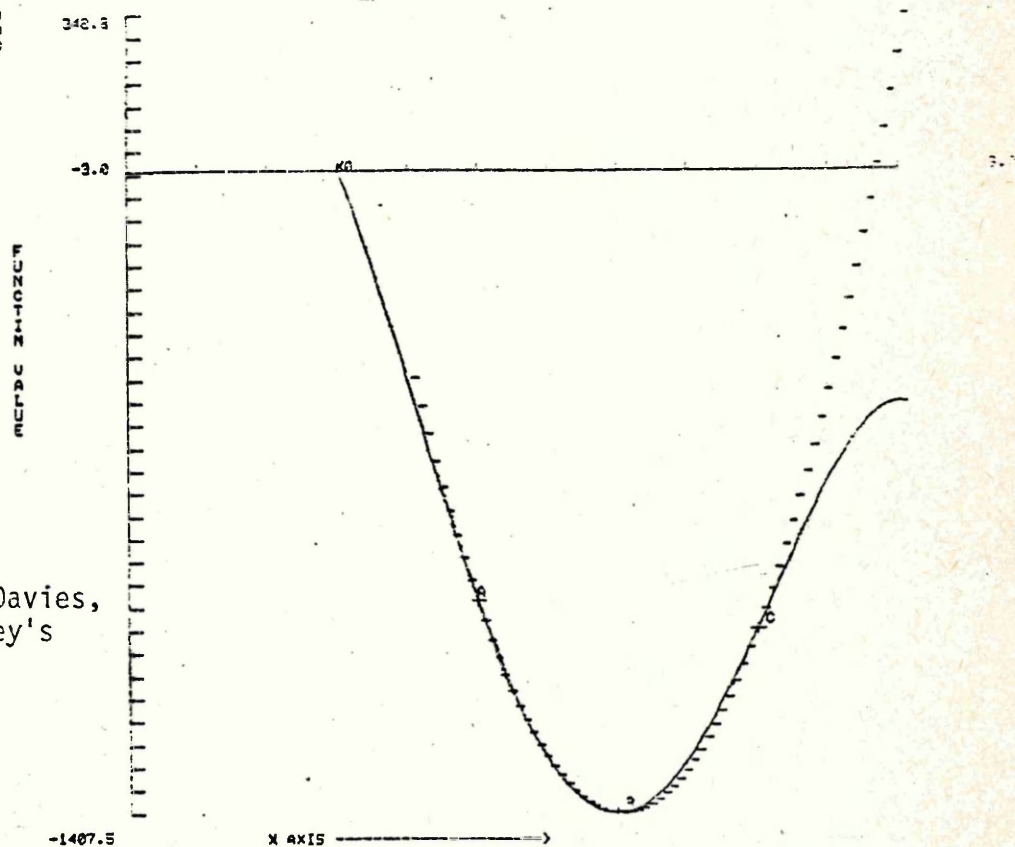


Fig. 5.10c
 Function (2)
 Two Stages of Davies,
 Swann and Campey's
 Method.

PRESS 1 TO FINISH
 PRESS 2 TO RUN
 PRESS 3 TO TEST NEW FUNCT.
 PRESS 4 TO CHANGE JINDCU
 PRESS 5 TO H. INTL. PT.

PRESS F FOR LINSO. TCH.
 PRESS G FOR OLS. TCH.
 PRESS H FOR TIGHT. TCH.

CORD. OF. A = 1.5-1381.0
 CORD. OF. B = 4.6-1408.3
 CORD. OF. C = 4.5-1373.3

MIN. REM. ED
 FRIM = 1406.353
 XMIN = 3.926

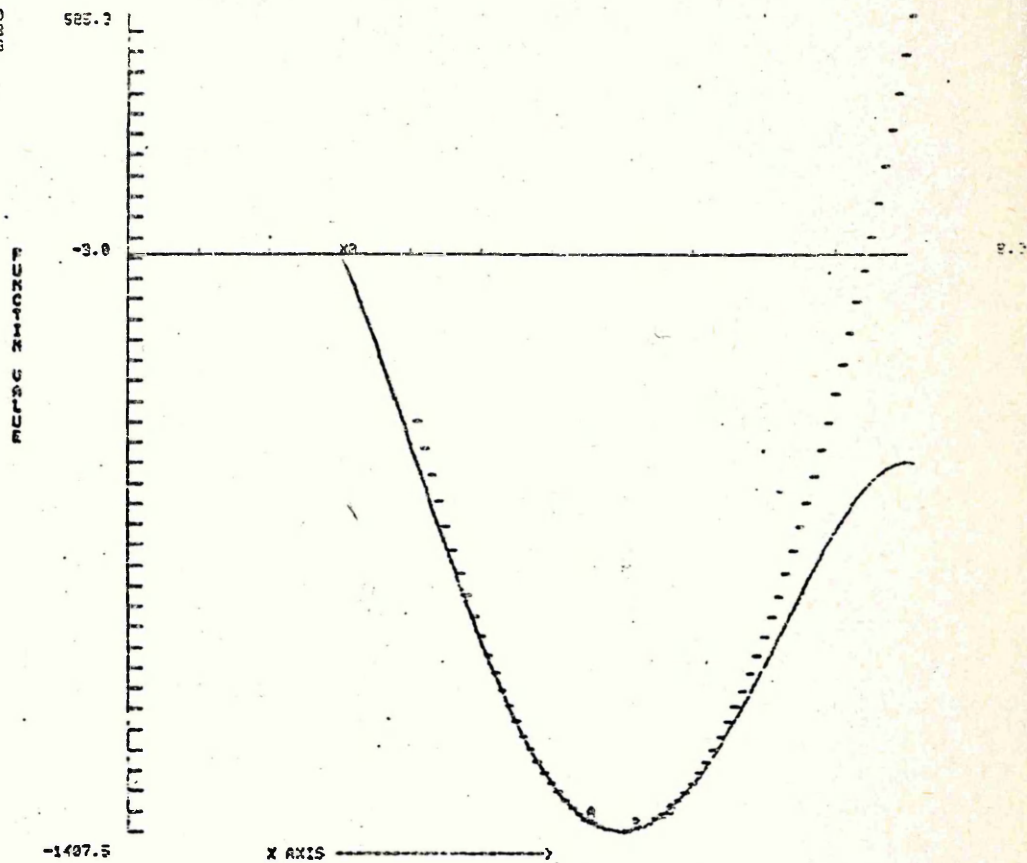
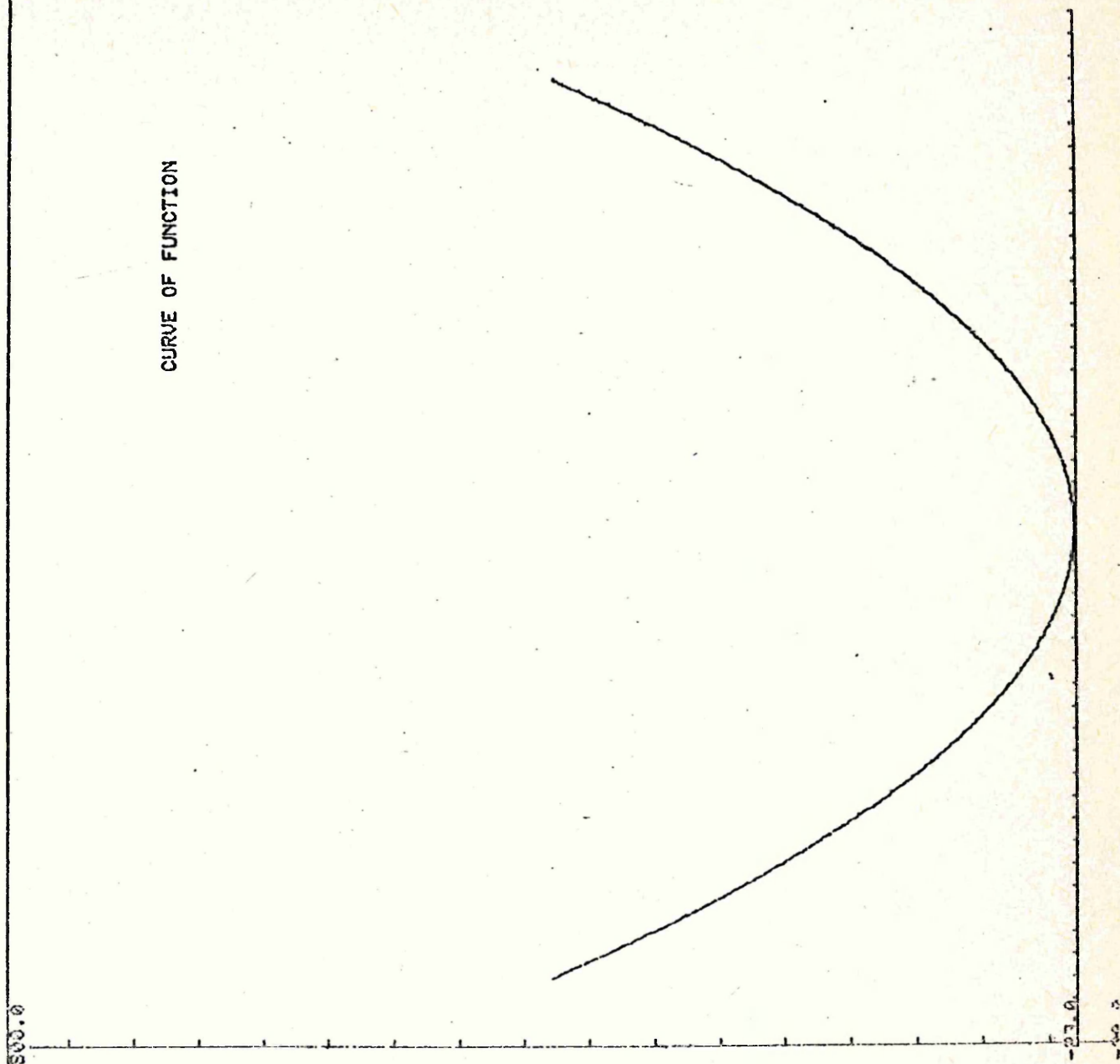


Fig. 5.10d Final Stage of Davies, Swann and Campey's Method.

PRESS C TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.
PRESS F FOR N. INTL. PT.
PRESS G TO TEST NEW FUNCT.
PRESS H TO CHANGE WINDOW
PRESS I TO FUNCT CURVE

CURVE OF FUNCTION

Fig. 5.11a
Function (3)



PRESS F FOR CURSOR
PRESS C TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

PRESS G FOR QUB.TCH.
PRESS H FOR QUAD. TCH.

A=C 6.250 X= 2.500
B=C 0.563 X= 0.750
A=C 0.016 X=-0.125
B=C 0.023 X= 0.313
B=C 0.029 X= 0.094
A=C 0.000 X=-0.016
B=C 0.002 X= 0.039
MIN REACHED
FMIN= 0.000
XMIN=-0.016

44.0

CURVE OF GRADIENT

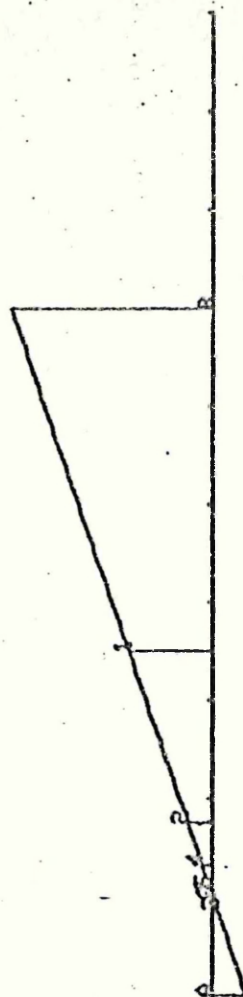


Fig. 5.11b

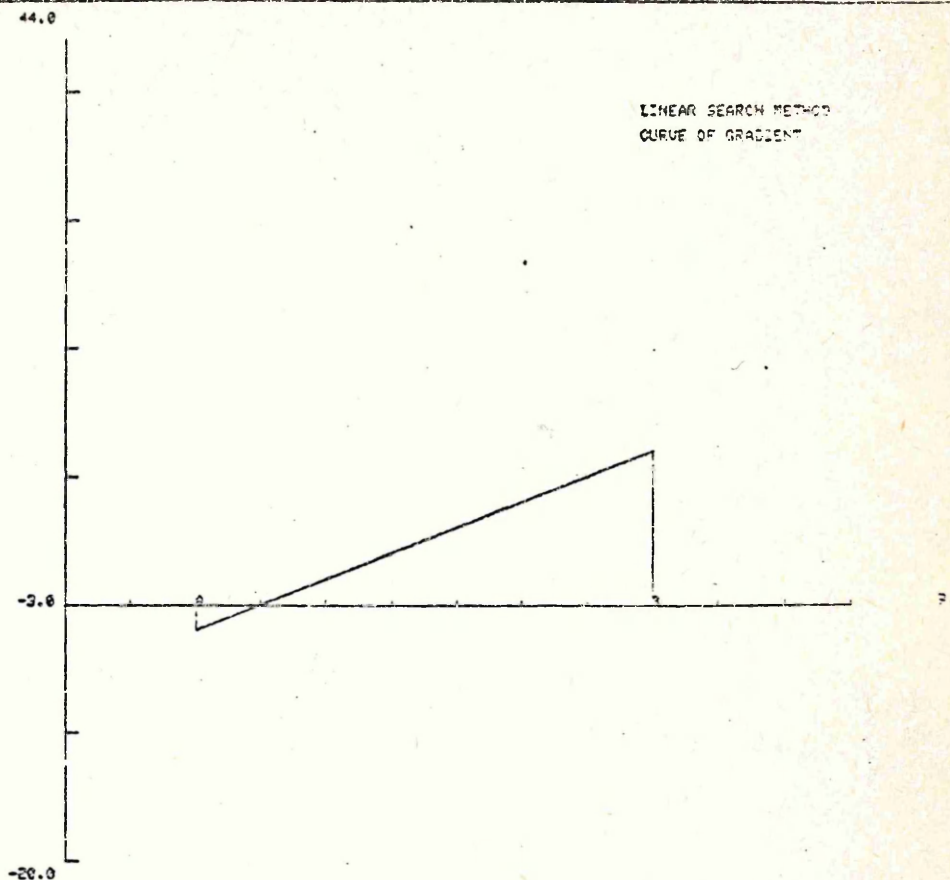
Function (3):
Midpoint Technique.

PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C TO TEST NEW FUNCT.
 PRESS D TO CHANGE WINDOW
 PRESS E FOR N. INTL. PT.

PRESS F FOR MIDP.TCH
 PRESS G FOR CUB.TCH.
 PRESS H FOR GLAD. TCH.
 PRESS I FOR PLOTION CURVE

MIN. REACHED AFTER 1 ITER.
 F= 0.0 X= 0.0

LINEAR SEARCH METHOD
 CURVE OF GRADIENT



PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C TO TEST NEW FUNCT.
 PRESS D TO CHANGE WINDOW
 PRESS E FOR N. INTL. PT.

PRESS F FOR MIDP.TCH
 PRESS G FOR CUB.TCH.
 PRESS I TO FUNCT CURVE

MIN. REACHED AFTER 1 ITER.
 F= 0.0 G= 2.0 X= 3.0

CURVE OF GRADIENT

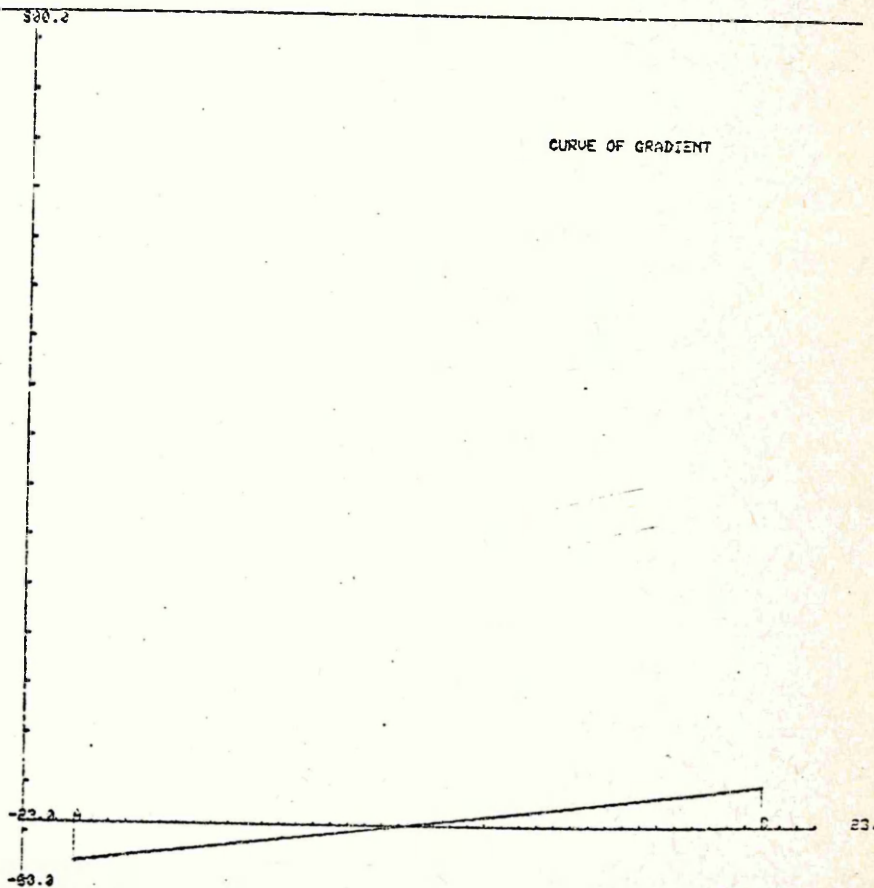


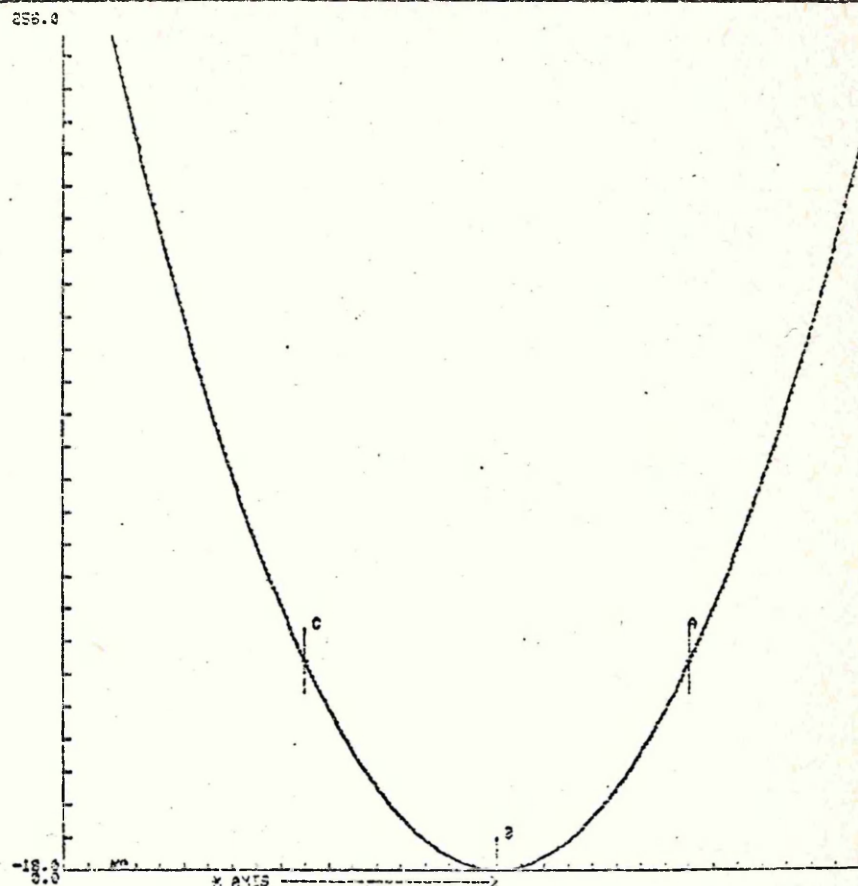
Fig. 5.1c
 Function (3)
 Linear Search Method.

PRESS B TO FINISH
 PRESS C FOR CURSOR
 PRESS D TO TEST NEW FUNCT.
 PRESS E TO CHANGE WINDOW
 PRESS F FOR N. INTL. PT.

PRESS F FOR LINSO. TCH.
 PRESS G FOR OUT. TCH.
 PRESS H FOR ADPT. TCH.

COORD. OF A* 8.0 64.0
 COORD. OF B* 8.0 0.0
 COORD. OF C* -8.0 -64.0
 PRESS C TO CONTINUE
 MIN. SEARCHED
 MIN* 3.000
 XMIN* -0.000

FUNCTION VALUE



COORD. OF A* 8.0 64.0
 COORD. OF B* 8.0 0.0
 COORD. OF C* -8.0 -64.0
 PRESS C TO CONTINUE

FUNCTION VALUE

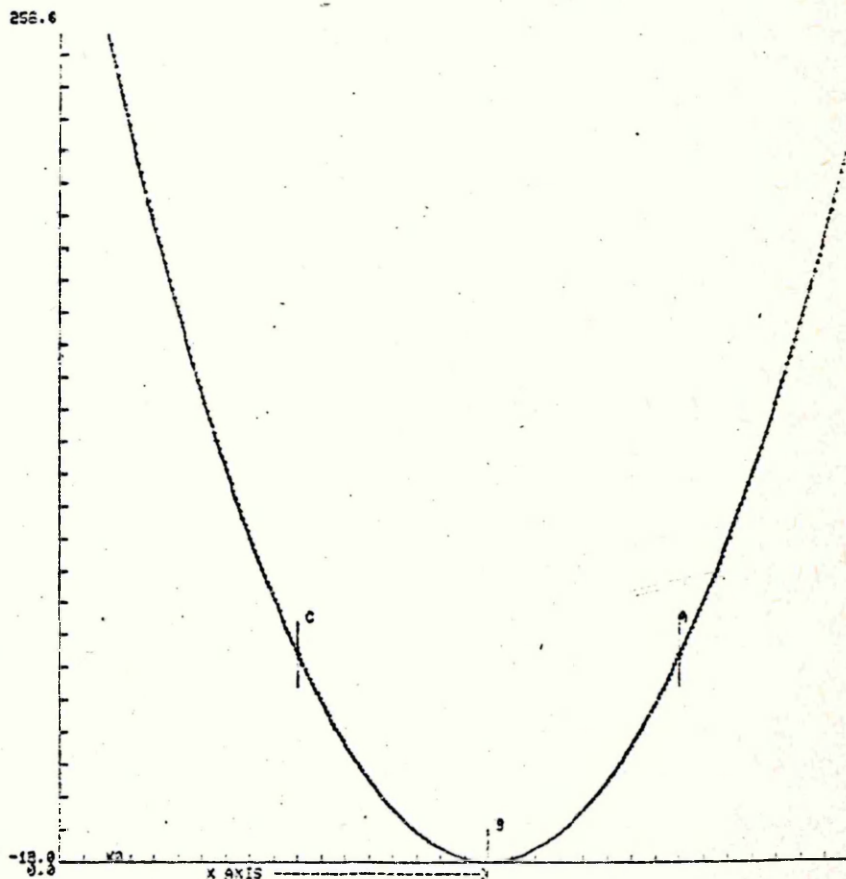


Fig. 5.11d
 Function (3)
 D.S.C. Method.

PRESS F TO FINISH
 PRESS G FOR CURSOR
 PRESS H TO TEST NEW FUNCT.
 PRESS J TO CHANGE WINDOW
 PRESS K FOR N. INTL. PT.

PRESS F FOR INSC. TCH.
 PRESS G FOR MIDP. TCH.
 PRESS H FOR QUAD. TCH.

ROOT OF CUBIC= 0.092
 MIN HAS REACHED
 FMIN= 0.007
 XMIN=-0.008

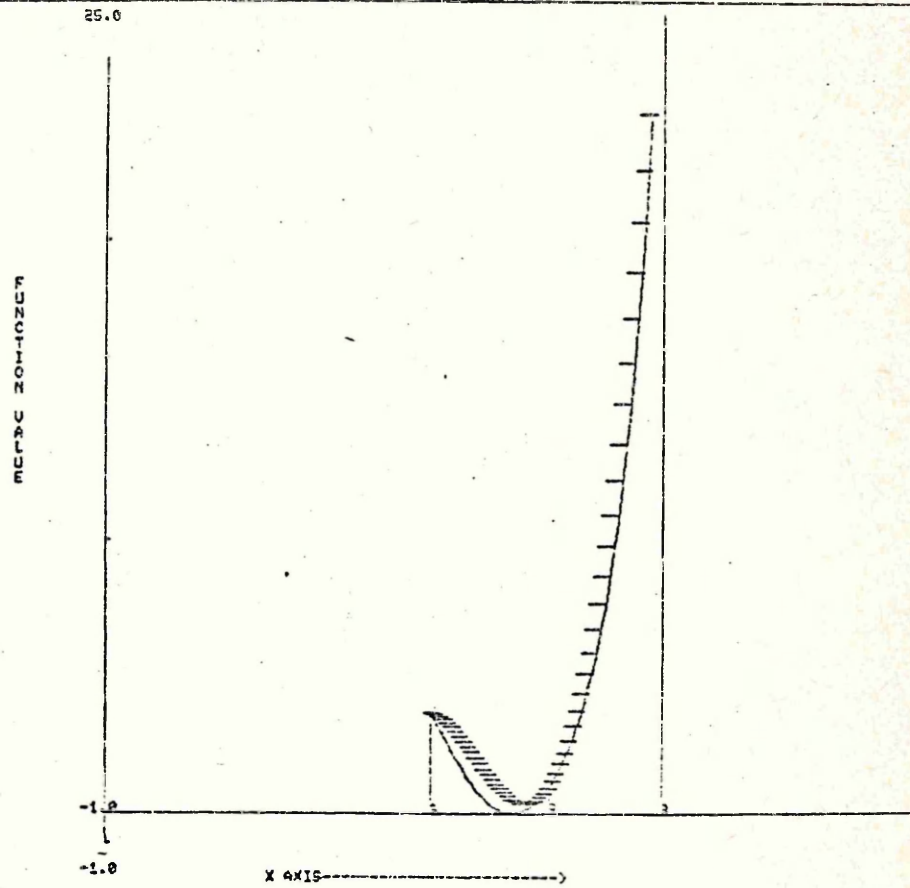


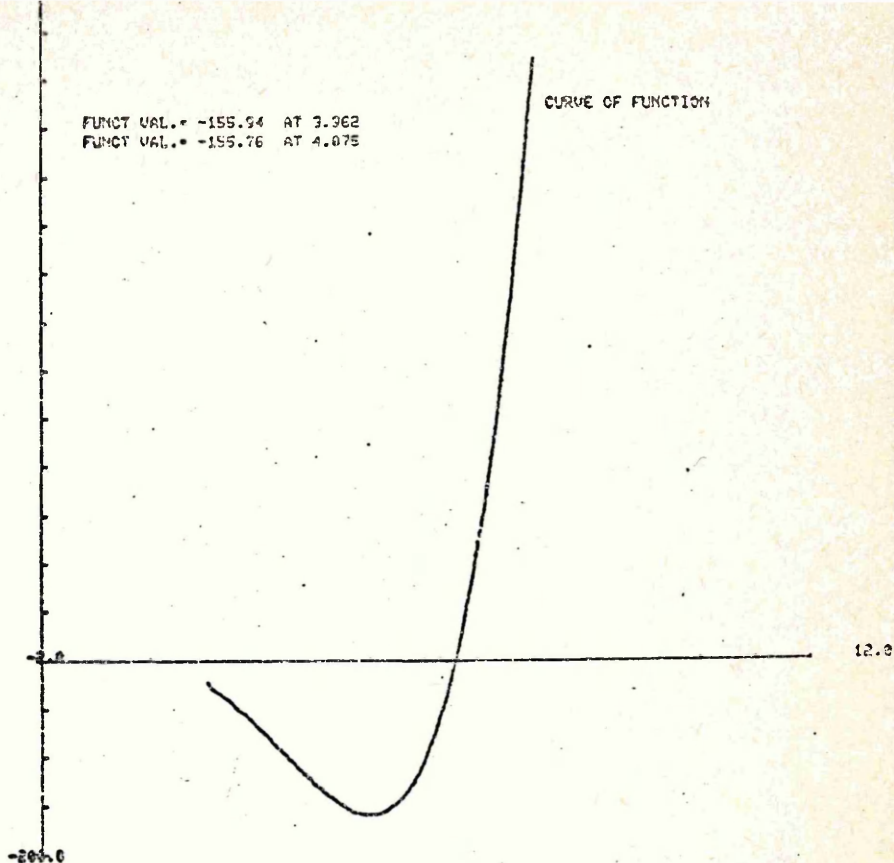
Fig. 5.11e Davidon's Cubic Method.

HIGHEST AND LOWEST PT. OF Y AXIS
 YS = 627.00 YA = -155.00
 WINDOW VALUES, REAL
 -3.0 12.0 -200.0 650.0

FUNCT VAL. = -155.54 AT 3.962
 FUNCT VAL. = -155.76 AT 4.075

CURVE OF FUNCTION

Fig. 5.12a
 Function (4)



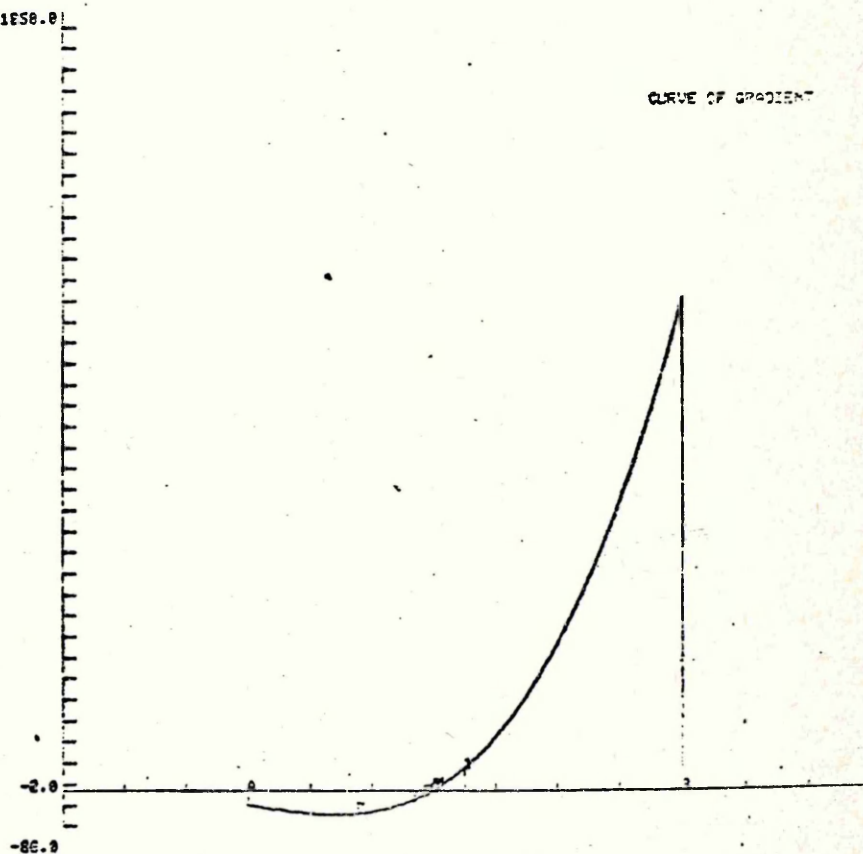
PRESS F TO FINISH
 PRESS F FOR CLASH
 PRESS F TO TEST NEW FUNCT.
 PRESS F TO CHANGE WINDOW
 PRESS F FOR N. INTL. PT.

PRESS F FOR LINSO. TCH.
 PRESS F FOR CL3. TCH.
 PRESS F FOR CLAD. TCH.

B-C
 F=-142.937 X= 4.500
 A-C
 F=-111.371 X= 2.750
 A-C
 F=-159.707 X= 3.625
 B-C
 F=-155.023 X= 4.051
 A-C
 F=-155.020 X= 3.344
 A-C
 F=-155.369 X= 3.353
 B-C
 F=-155.557 X= 4.008
 MIN REACHED
 FMIN= -155.963
 XMIN= 3.553

CURVE OF GRADIENT

Fig. 5.12b
 Midpoint
 Technique.



PRESS C TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

PRESS B FOR COS. CH.
PRESS H FOR QUAD. TCH.
PRESS I FOR FUNCTION CURVE

714.0

VALS. AFTER 10 EVALS.
F = -155.93 X = 3.976

MIN. REACHED AFTER 17 ITER.
F = -156.00 X = 4.060

LINEAR SEARCH METHOD
CURVE OF GRADIENT

Fig. 5.12c
Linear Search Method.

-76.0

PRESS A TO FINISH
PRESS B TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

PRESS F FOR LINSO. TCH.
PRESS G FOR QUB. TCH.
PRESS H FOR ADPT. TCH.

2342.4

COORD. OF A = 0.00 -56.9
COORD. OF B = 4.06 -143.9
COORD. OF C = 0.00 330.1
PRESS C TO CONTINUE
MIN. REACHED
FMIN = -155.933
XTMIN = 4.048

FUNCTION MINIMUM

Fig. 5.12d
Davies, Swann and
Campey's Method.

-153.0

X AXIS

PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C TO TEST NEW FUNCT.
 PRESS D TO CHANGE WINDOW
 PRESS E FOR N. INTL. PT.

PRESS F FOR LINSO. TCH.
 PRESS G FOR MIDP TCH.
 PRESS H FOR QUAD. TCH.

ROOT OF CUBIC= 4.000
 MIN HAS REACHED
 FMIN=-156.000
 XMIN= 4.000

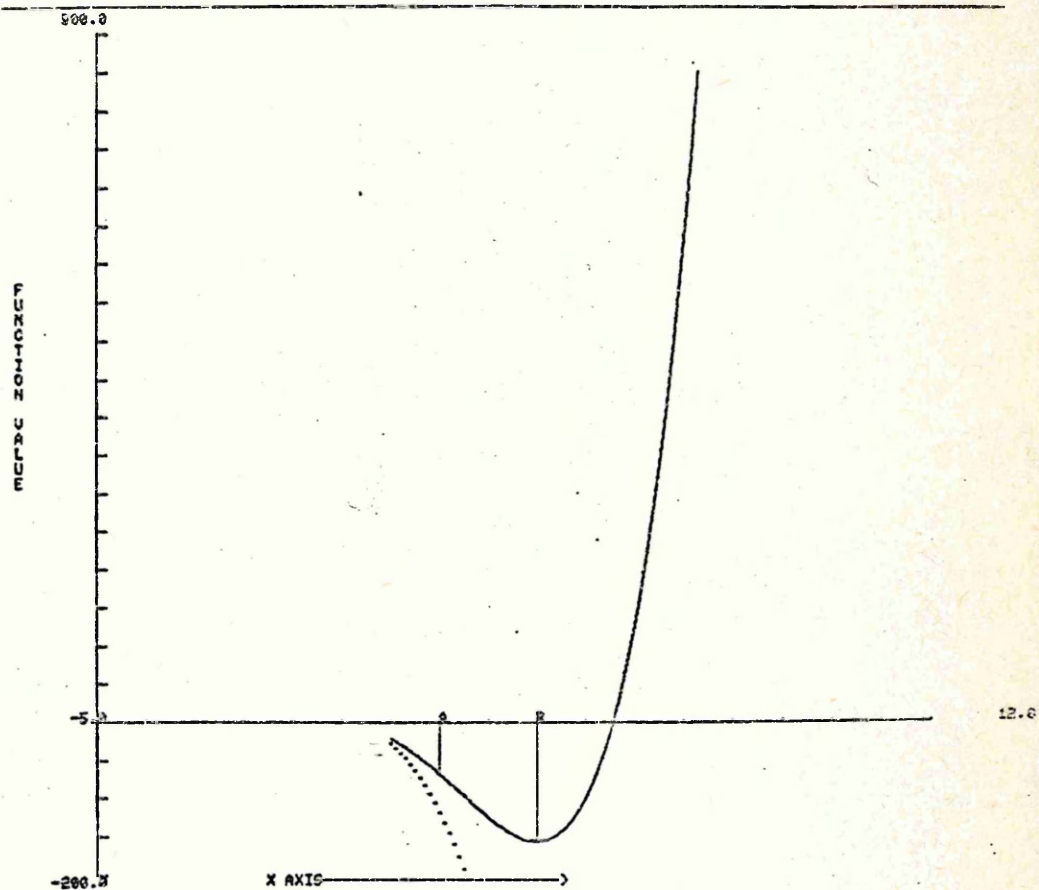


Fig. 5.12e

Fig. 5.12e Davidon's Cubic Method.

PRESS A TO FINISH
PRESS B FOR CURSOR
PRESS C TO TEST NEW FUNCT.
PRESS D TO CHANGE WINDOW
PRESS E FOR N. INTL. PT.

PRESS F FOR LINSO. TCH.
PRESS G FOR QUS. TCH.
PRESS H FOR QUAD. TCH.
PRESS I TO FUNCT. CURVE

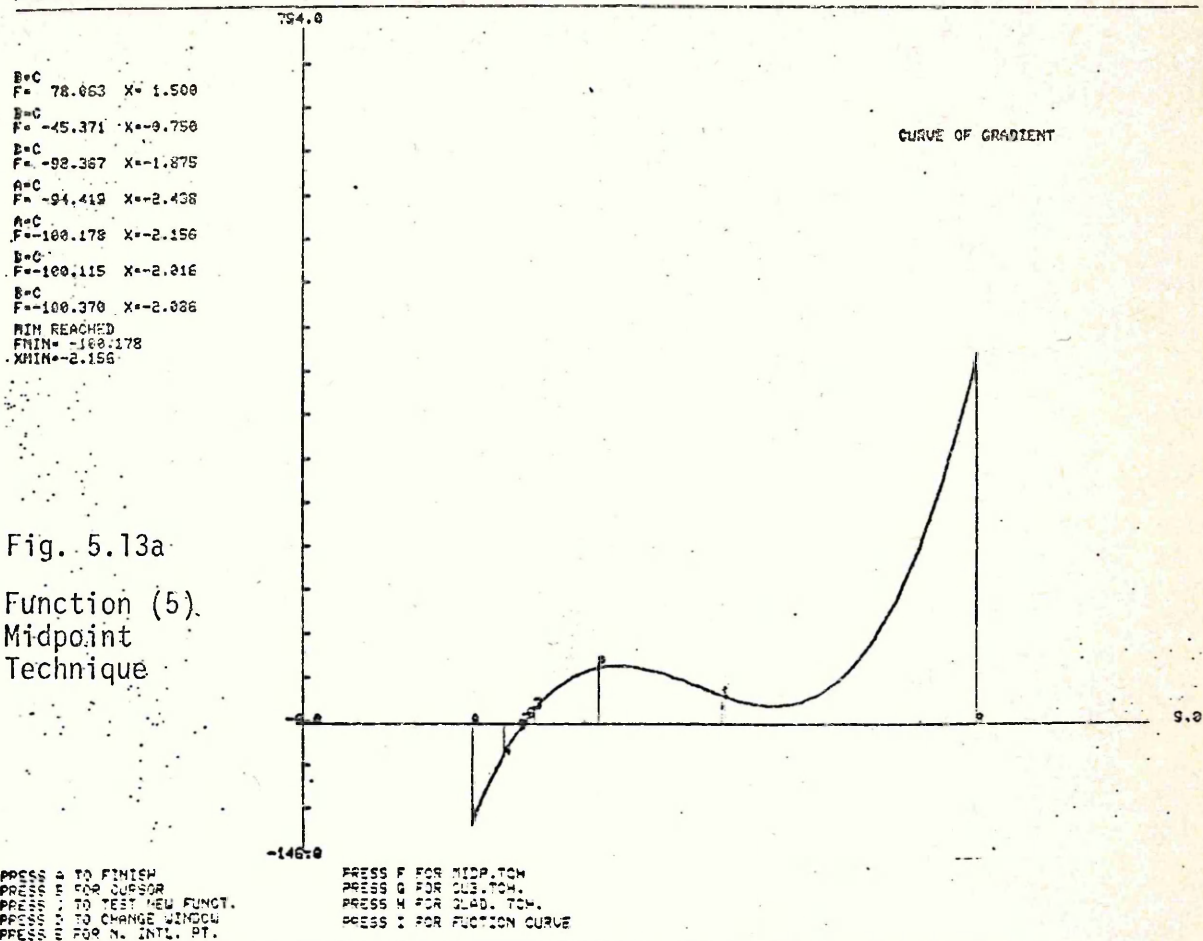


Fig. 5.13a

Function (5)
Midpoint
Technique

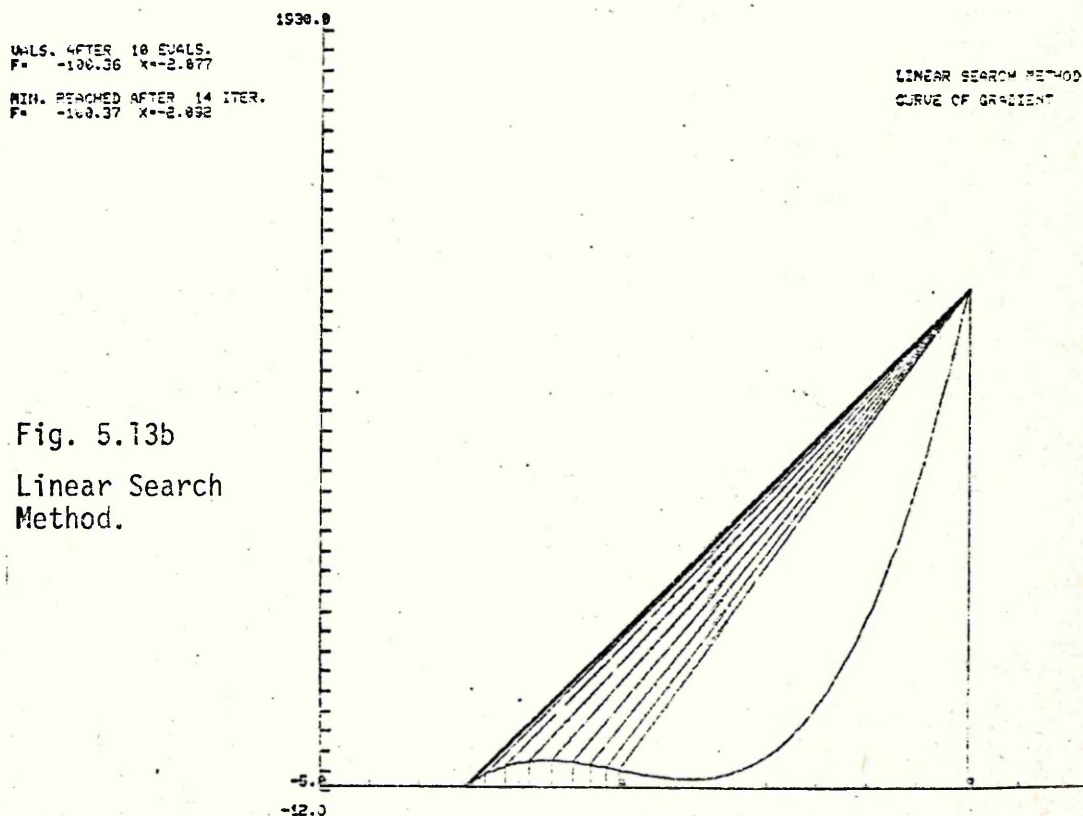
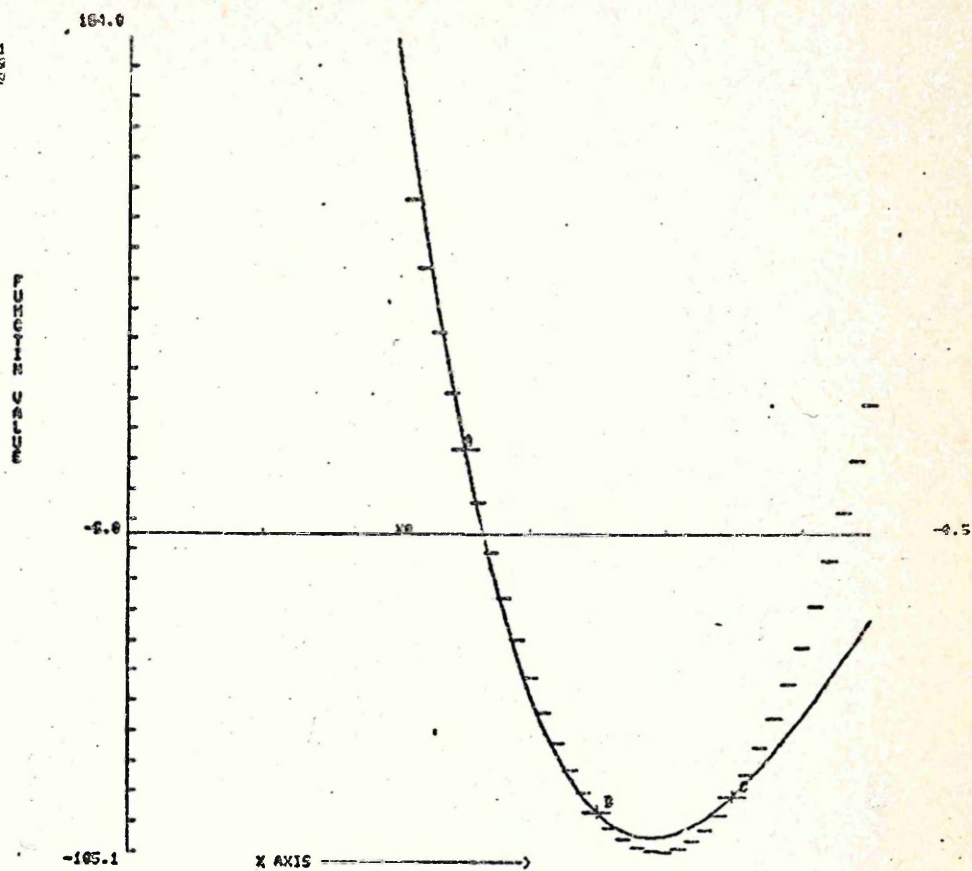


Fig. 5.13b

Linear Search
Method.

CORD. OF. A = -3.5 22.1
 CORD. PT. B = -2.5 -91.9
 CORD. PT. C = -1.5 -86.9
 PRESS C TO CONTINUE



PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C TO TEST NEW FUNCT.
 PRESS D TO CHANGE WINDOW
 PRESS E FOR N. INTL. PT.

PRESS F FOR LINES. TCH.
 PRESS G FOR CUR. TCH.
 PRESS H FOR MIDPT. TCH.

CORD. OF. A = -3.5 22.1
 CORD. PT. B = -2.5 -91.9
 CORD. PT. C = -1.5 -86.9
 PRESS C TO CONTINUE
 MIN. REACHED
 MIN = -108.358
 XMIN = -2.076

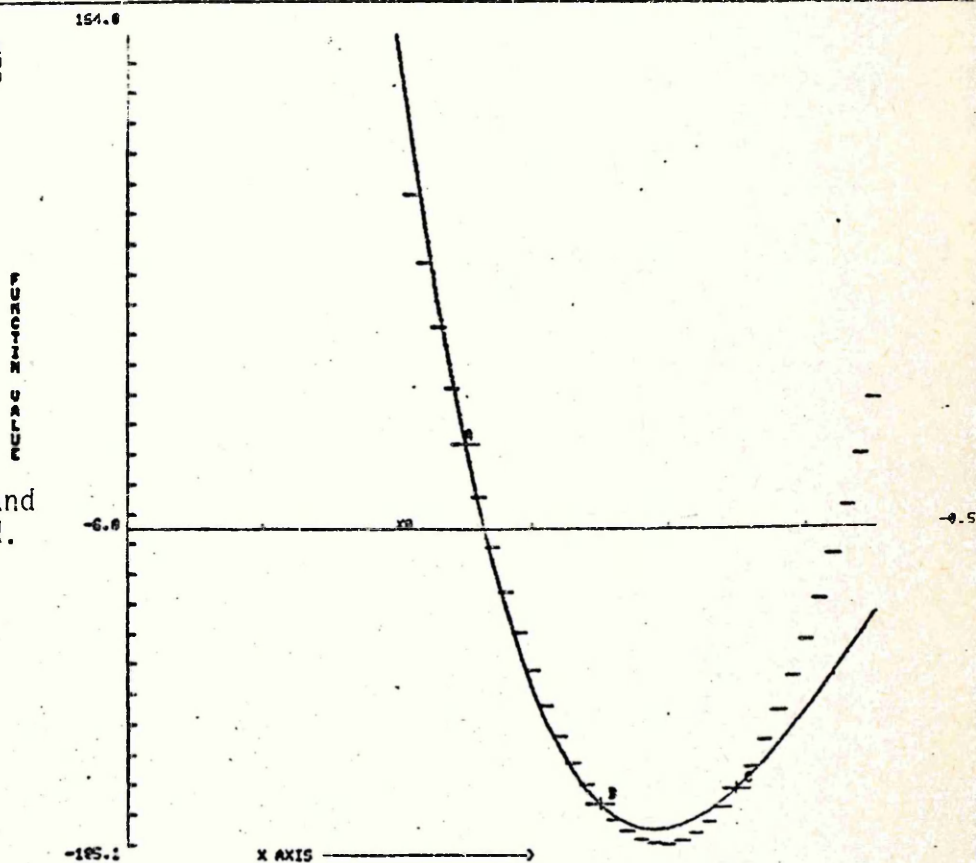
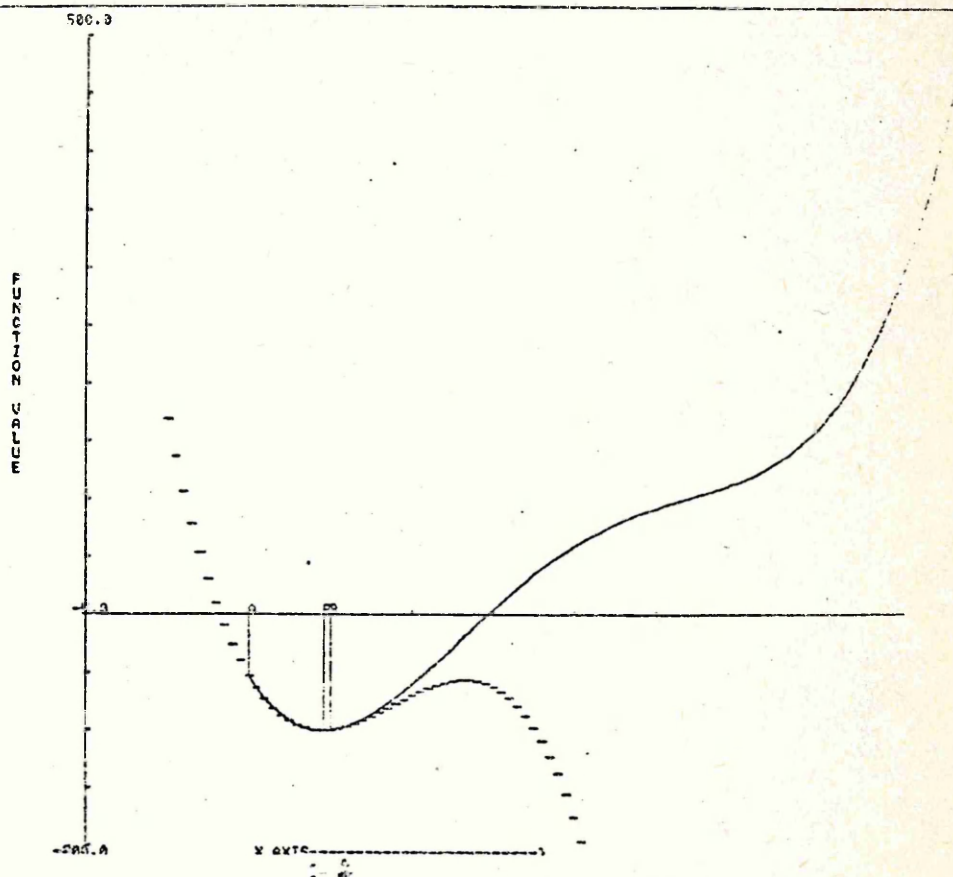


Fig. 5.13c
 Davies, Swann and
 Campey's Method.

PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C TO TEST NEW FUNCT.
 PRESS D TO CHANGE WINDOW
 PRESS E FOR N. INTL. PT.

PRESS F FOR LINSO. TON.
 PRESS G FOR PLOT TON.
 PRESS H FOR CLAD. TON.

ROOT OF CUBIC: -2.893
 MIN HAS REACHED
 FMIN=-100.371
 XMIN=-2.893



ROOT OF CUBIC: -2.893
 MIN HAS REACHED
 FMIN=-100.371
 XMIN=-2.893

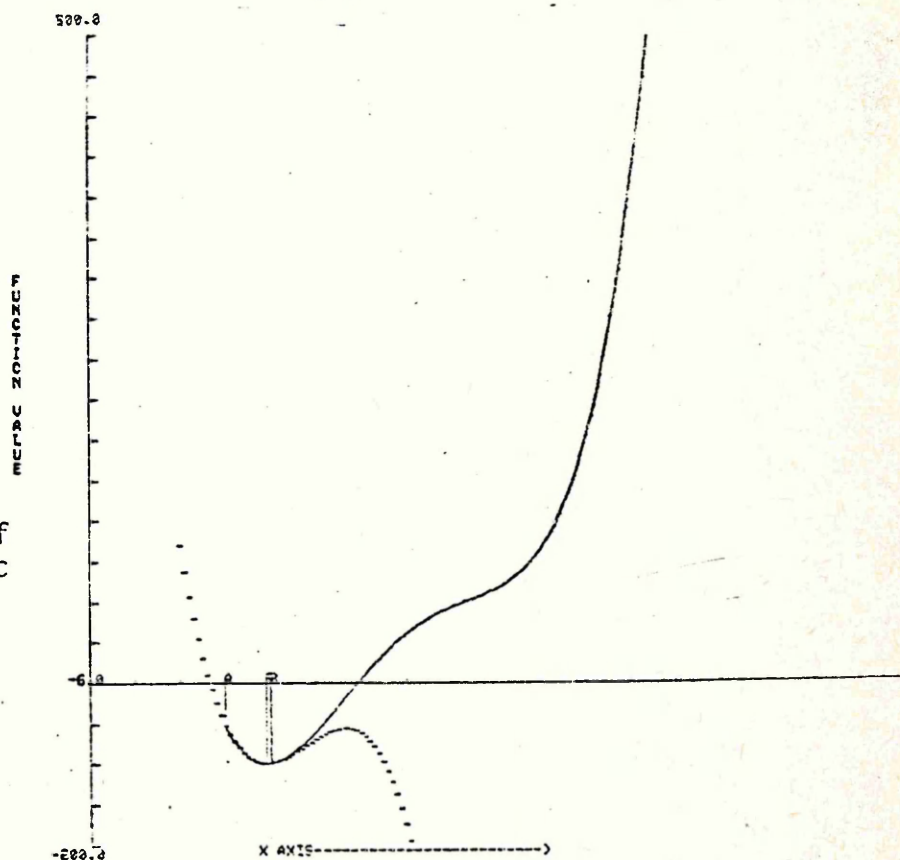


Fig. 5.13d
 Two Displays of
 Davidson's Cubic
 Search Method.

and Simplex. These two methods and their application to a two-dimensional function will be discussed in the later sections. However, the approach which is used here is based on drawing the contour of the function and finding its minimum visually.

5.2.1 Contour plotting routine

There are several methods for plotting contours, namely Dayhoff(38), Robinson and Scarton (39), Cottafava and Le Moli (40) and Crane (41). These methods are based on the evaluation of exact roots along the sides of rectangles and the generation of one or two straight lines to join them. McLain (42) and Sutcliffe (43) use algorithms which are based on subdivision of the rectangle into a grid of smaller rectangles and the generation of many of straight lines so that apparently continuous smooth curves join the roots along the side of the rectangle.

It is an adaptation of the last method which is used in the work of this thesis, the original, ALGOL procedures having been translated to FORTRAN for the purpose.

The program starts by asking the user to input certain initial values. These include the number of variables, the step size, window range and the value for x_{\min} , x_{\max} , y_{\min} and y_{\max} . If the function does have any constraints the user must specify the types of constraints and the number of each type.

The area between x_{\min} , x_{\max} , y_{\min} and y_{\max} is divided into four quadrants and since tracing the contour curve is the same for all four quadrants the procedure is only described for one quadrant.

Having defined the co-ordinates of the corner points of the quadrant the function value is found at points $(x_A, y_A), (x_B, y_A)$ where x_A and y_A are the points where the x axis and y axis have to

be divided, $x_B = x_{\max}$. If the two function values have the same sign, it indicates that there are no roots, so the function values at the other corner points (x_A, y_B) and (x_B, y_B) are calculated and their signs are compared. If there is no change of sign in the function value on any of the four sides of the rectangle, then the search for a root must be directed to the next quadrant because this shows that the contour does not pass through the first quadrant. The same process is carried out for each quadrant until a root is found.

If a sign change is detected, then for the given interval length the number of intervals along the side is calculated. The midpoint of the side, rounded to an integer number of intervals is found. The function value at this point is calculated. The sign of this function determines which half the root lies in. This procedure is now applied to this half line and is repeated until the length is equal to that of an interval. This process allows a relatively precise location of the root on the interval length of the chosen side. This interval is used as the starting base line for the small rectangles into which the quadrant is being divided. The sign of the function is determined at the other two corners. The sign change along a side indicates that the curve will proceed from this side and this forms the base line for the next rectangle. This process is repeated until the edge of the quadrant is reached and the exit point is then recorded. The other corners are in turn investigated for a root and if the root is not in the exit list, then the above procedure is repeated. Otherwise, the other quadrants are examined to trace the curve.

5.2.2 Application of contour plotting routine to a two-dimensional problem

The contour plotting routine is designed to plot the smallest contour for the specified region, unless it is instructed otherwise by the user. This has proved to be useful especially for two-dimensional functions whose minimum can be visually located within the plot of a single contour. However, if the user is not satisfied with the minimum found in the first contour, the minimum values may then be used to specify a new region within which another contour is smaller than the previous value. It can therefore be concluded that for a two-dimensional function it is in principle possible to locate a minimum by simply reducing the size of the contour and locating the minimum visually each time.

Figure 5.14 (a) shows the contour of the two-dimensional Rosenbrock function and its approximate minimum at the second point. In this case the user has chosen the size of the contour. The co-ordinates of the second point have been used to define a new region for which a second contour is drawn. This is illustrated in figure 5.14 (b).

Figure 5.15 shows a close up of the second contour and as it can be seen the minimum has been visually located which is an acceptable value.

5.3 Problems in N-dimensions

The methods in general use for finding the optimum of a function of N-variables can be divided into two categories:-

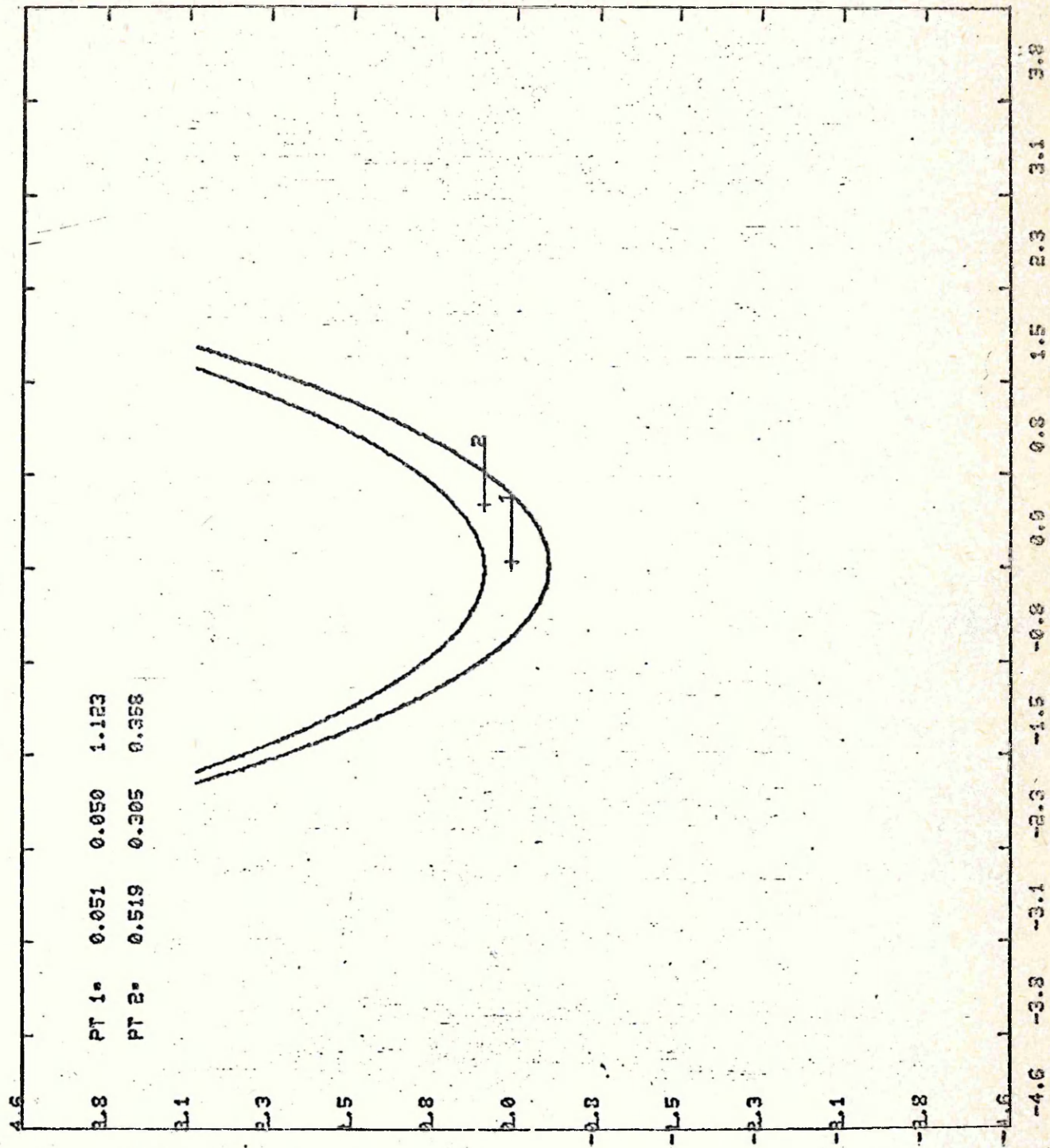
THIS PROBLEM DOES NOT HAVE
ANY CONSTRAINTS

DO YOU WISH TO INPUT CONTOUR
VARIABLES? IF NOT THE PROG.
WILL DECIDE FOR YOU
Y
CONTOUR VARIABLE, REAL
10.0

Fig. 5.14a

Contour of the
Rosenbrock Function.

FMIN= 1.000
FNC.EVAL. KN1= 3918



THIS PROBLEM DOES NOT HAVE
ANY CONSTRAINTS

DO YOU WISH TO INPUT CONTOUR
VARIABLES IF NOT THE PROG.
WILL DECIDE FOR YOU

Y
CONTOUR VARIABLE, REAL
10.0
RANGE OF X, REAL
0.519 3.0 -3.0 0.519
RANGE OF Y, REAL
0.305 3.0 -3.0 0.305

FMIN= 1.000
FNC.EVAL. KN1= 3918
FMIN= 0.358
NC.EVAL. KN1= 3152

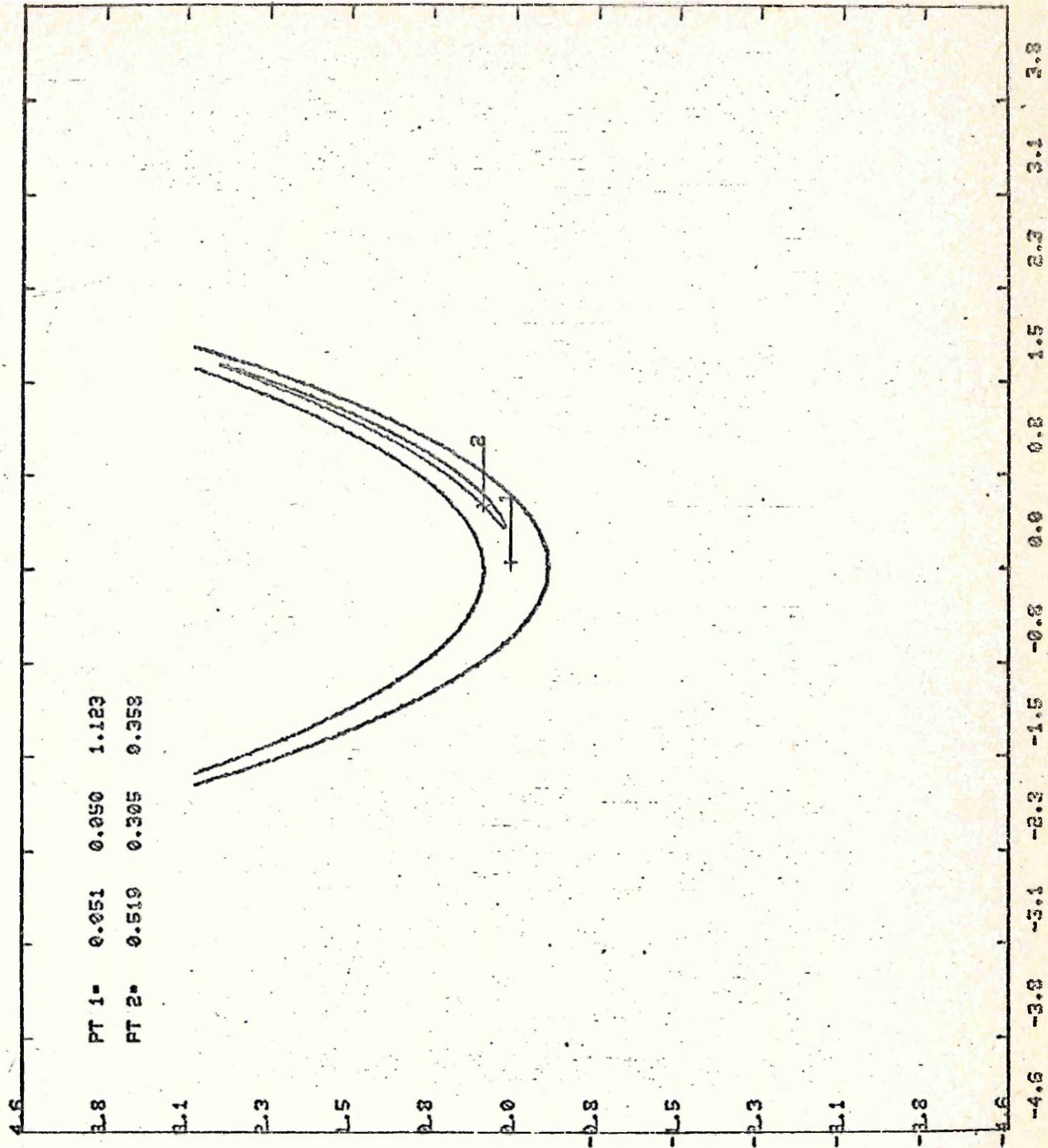


Fig. 5.14b

PRESS A TO FINISH
 PRESS E FOR CURSOR
 PRESS C FOR F.P.METHOD
 PRESS D FOR MIN COOR. FINDING
 PRESS E,F FOR 1,2ND CONTOUR

PRESS G FOR NEW PAGE & WINDO
 PRESS H FOR NEW PAGE & METHOD
 PRESS I STORE MIN.COOR. OF CONTOUR
 PRESS J IF ONLY A NEW PAGE IS NEEDED
 PRESS K TO CHANGE THE WINDO.

PRESS L TO CHANGE THE STEP SIZE
 PRESS ANY OTHER KEY FOR START

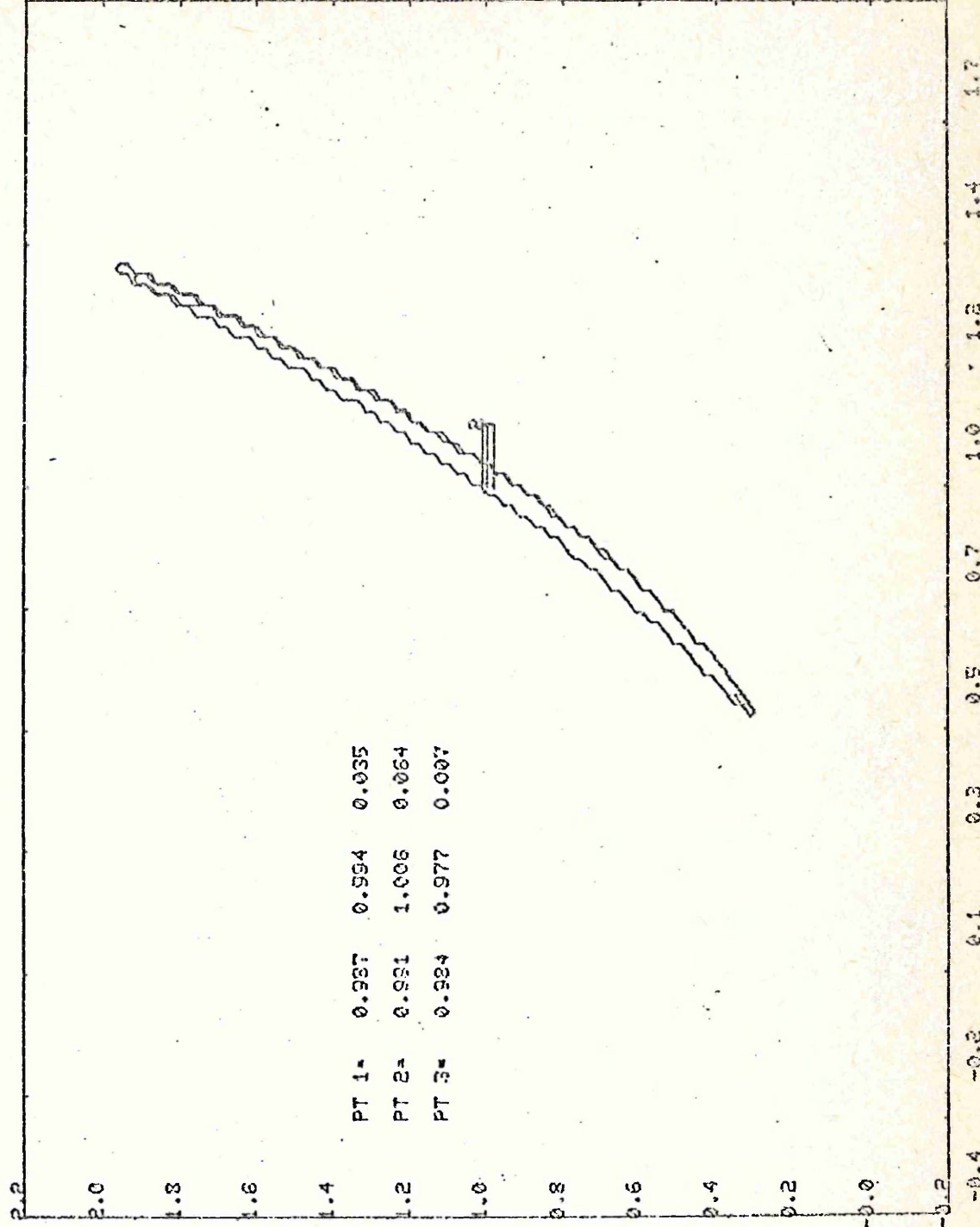
WITH CURSOR,A...I,AS BEFORE

THIS PROBLEM DOES NOT HAVE
 ANY CONSTRAINTS

Fig. 5.15

Min. of the Rosenbrock
 Function Located
 Visually.

FTMIN= 0.103
 FNC.EVAL. KN1= 1112



- (i) methods which use derivatives of the function,
- (ii) methods which do not require derivatives of the function.

In this thesis one method from each category has been chosen to be compared with a direct search graphical analysis procedure which uses contours of an N-dimensional function in a two-dimensional space. The two conventional methods are:-

- (1) Fletcher and Reeves method
- (2) Simplex search method.

The above two methods and the graphical method are combined to produce the second part of the system which consist of a suite of N-dimensional optimization methods.

5.3.1 A program for N-dimensional problems

The basic structure of this program is similar to that of the one-dimensional program providing the same range of facilities for the user. For example, the user can test more than one function with more than one method in any session.

When entering this program an initial display frame is presented to the user. This frame consists of the initial instructions and a list of available methods.

After each method is terminated a list of options is displayed on the screen. These options can be divided (as before) into two groups. The 'internal' options and the 'external' options. The number of options available for both types differ for each method and a list of them will be given when each is described.

Figure 5.16 shows the flow diagram of the N-dimensional program.

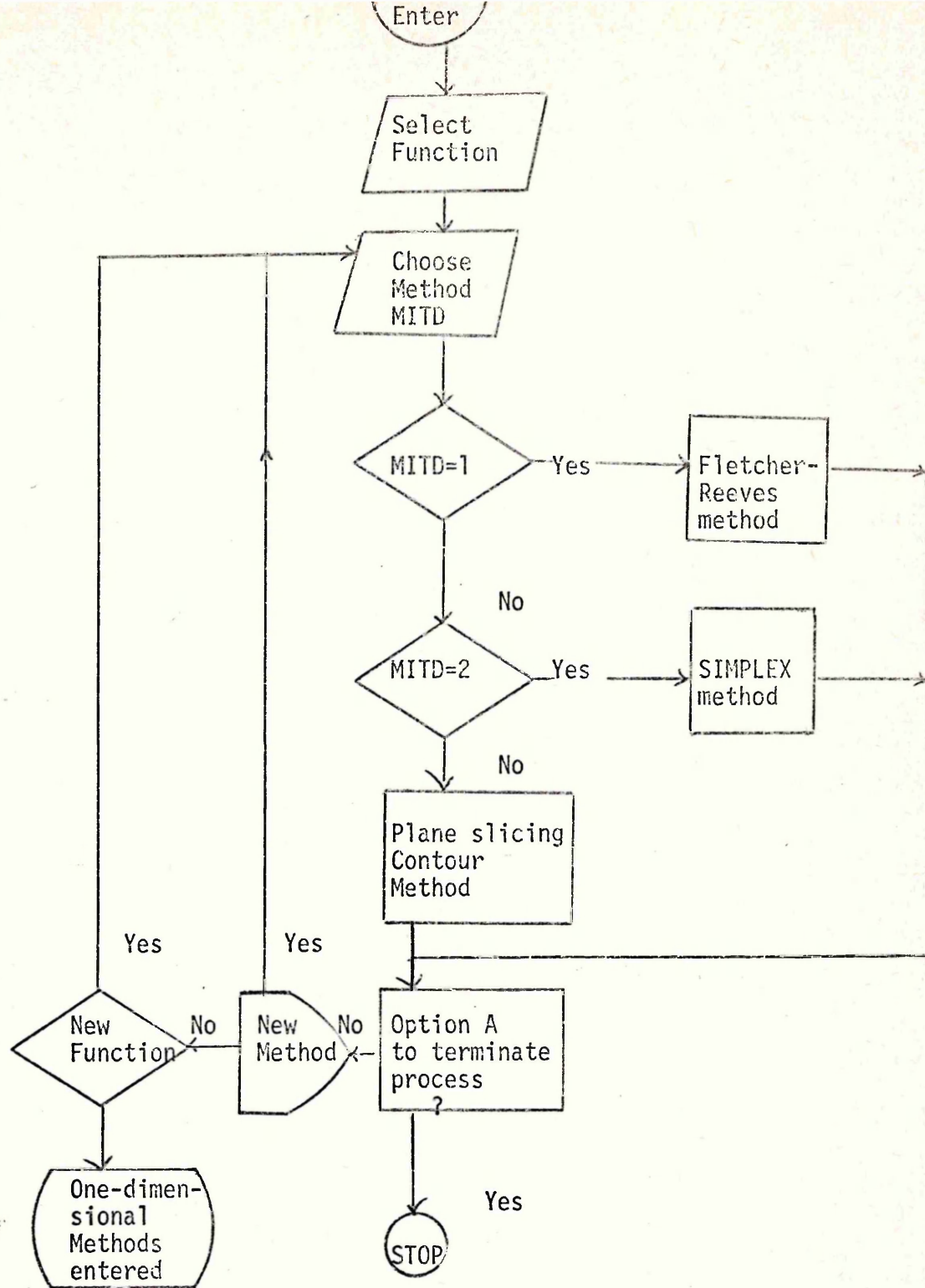


Fig. 5.16 Flow Diagram of N-dimensional Program.

5.3.2 Method of Fletcher and Reeves

This is a quadratically convergent method and was first introduced in 1964. It makes use of conjugate gradient vectors.

This method is applied to functions of N-variables whose function value of $f(x_i)$ and gradient vector $g(x_i)$ can be evaluated at any general point x_i , ..., $i = 1, \dots, N$. The basic algorithm is;

x_0 = starting point

$g_0 = g(x_0)$, $P_0 = g_0$

x_{i+1} = point of minimum of $f(x)$ on the line through x_i and in the direction of P_i .

Cubic interpolation is used in the linear search part of the algorithm. It has been found that the cubic interpolation technique, when applied to certain test functions, predicts a minimum t_m close to one of the bounds so that the range becomes effectively constant. Dixon (44) has introduced a safeguarded method which requires at least one more function evaluation at each iteration. This approach cuts down the range in one cycle rather than having to reduce the range several times.

The method is terminated when one of the two conditions is satisfied:

- (1) For a point x_i to be minimum, the value g_i^2 must be zero or within sufficiently small tolerance criterion.
- (2) The iterations are continued until a complete cycle of $(N+1)$ iterations has produced no reduction in the value of the function.

A further safeguard to save unnecessary wastage of computer time is to apply some arbitrary limit to the number of $(N+1)$ iterations allowed.

The Fletcher - Reeves method is one of the most efficient algorithms utilising conjugate gradients. This method has also been embodied in procedures for constrained optimization because of its modest memory requirements - it only requires two vectors to be stored. The last factor has influenced the choice of the method for this thesis.

The FORTRAN subroutine used here is based on the ALGOL procedure produced by Fletcher & Reeves (28). The program has been modified to suit the computer graphics user. The program now provides the user with a certain degree of interaction and control over the running of the algorithm.

Initially, the number of variables, their starting values, and the estimate of the value of the function are input. The user must then define the values for two limits. The first limit is on the number of cubic interpolations performed after each linear search. This limit gives an indication of how the interpolation is proceeding. When the number of interpolation reaches the limit a message is displayed on the screen which allows the user either to proceed with the interpolation with a new limit, or to choose one of the available options which will be displayed on request.

The second limit is for the number of iterations performed in search of the minimum. The procedure is terminated when the number of iterations exceeds the limit, indicating that the function has not converged with the given initial values. This is followed by a display of options from which the user can decide on the next step, i.e. change the initial value, change the method, or test a new function etc.

At the start of the procedure the user can ask for a display of the linear search and the approximate curve. The display of this curve is under user's control.

After the display of the linear search and the approximate curve is completed, the cursor can be employed to locate the minimum, t_e , visually. The user can then over-ride the calculation of t_e and input the minimum value which has been located visually. The user has this choice every time a linear search is performed.

To provide the user with more information about the performance of the method a minimization path in any two dimensions could also be displayed once the procedure is terminated.

For difficult functions, where slow convergence has been encountered, the graphical presentation of the function may prove to be useful in giving an insight into how the method is working. The rate of convergence could also be speeded up by the input of estimates treated visually in some cases.

Figures 5.17 (a) and 5.17 (b) show the steps used by this method to find the minimum value of the Rosenbrock function.

Figure 5.17 (a) shows the pattern of the linear search. The actual values of the points on the graph are shown on the left hand side of the screen. The pause at this stage allows the user to get a hard copy of the display before proceeding to the next stage by pressing the key F.

Figure 5.17 (b) shows the approximate curve and the function value after the cubic interpolation. The available options are also displayed and the user can now choose the option 'B' for cursor investigation of a few points on the curve and compare these values

with the ones found by cubic interpolation. If none produce better results, then option 'D' uses the cubic minimum to end the linear search. Otherwise option 'C' replaces the cubic minimum with the one selected by the user. In the case illustrated however, the cursor has not been used and the user has chosen to accept the cubic minimum and continue with the minimization process which finds the minimum after thirty three iterations.

The user can also request to see the minimization path. This path can also be drawn after a contour of the function has been plotted. Figure 5.18 shows the minimization path of the Rosenbrock function and its contours.

Figure 5.19 shows the flow diagram of the Fletcher and Reeves method.

5.3.3 Simplex method

This method involves placing observations at the vertices of an equilateral triangle and evaluating the function at each point.

A simplex in N-dimensions consists of a pattern of at least (N+1) points denoted by x_i , $i = 1, \dots, N+1$. The initial 'Simplex' is normally found by making an estimate x_1 and then taking a step along each axis. At any one stage the values F_i , $i = 1, \dots, N+1$ of the function at the points x_i are stored. The values F_i are compared to determine,

$$F(x_H) = F_H = \max_i F_i \quad i = 1, \dots, N+1$$

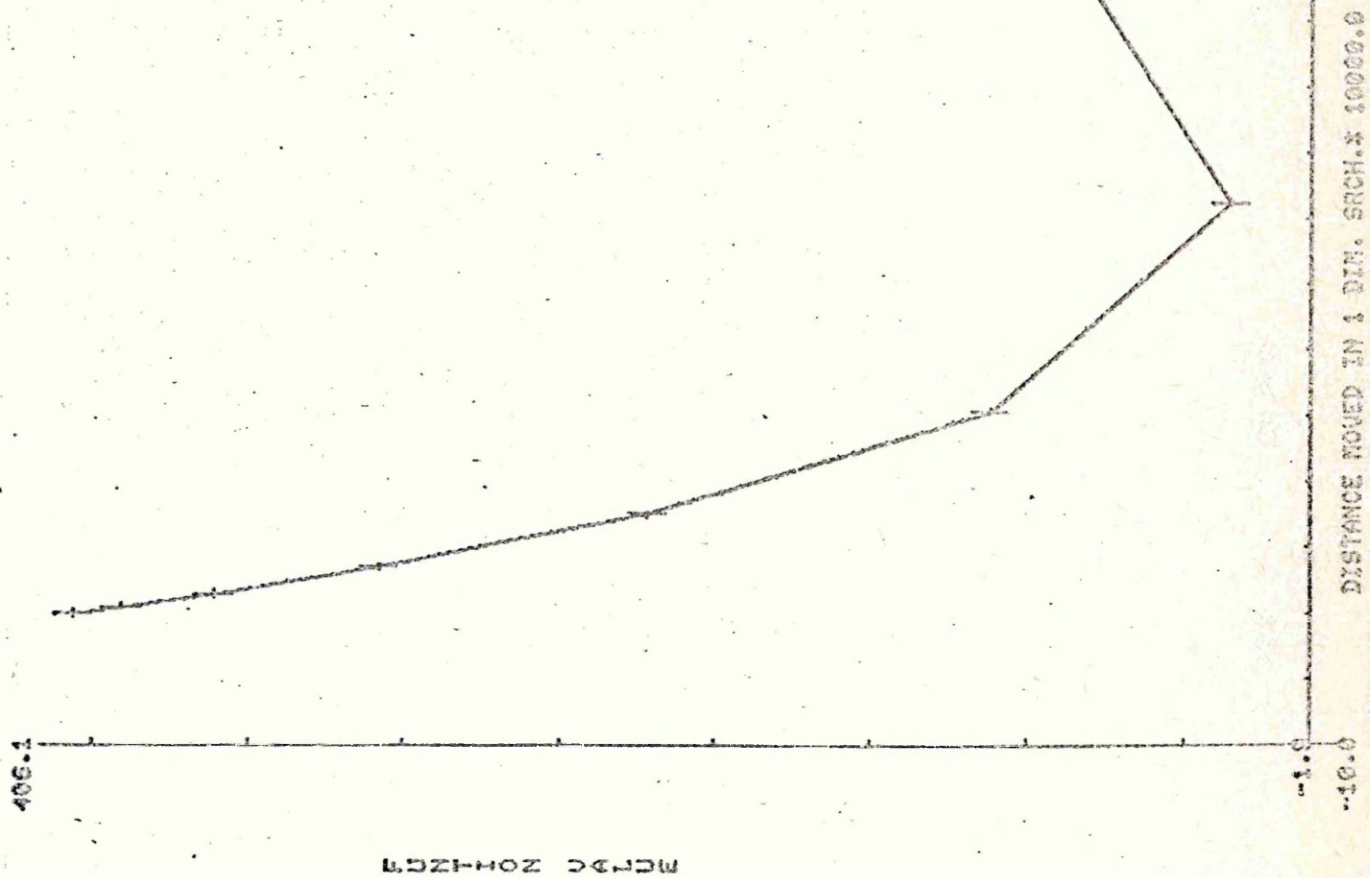
$$F(x_G) = F_G = \max_i F_i \quad i = 1, \dots, N+1, \quad i \neq H$$

$$F(x_i) = F_i = \min_i F_i \quad i = 1, \dots, N+1$$

PT. 1
(0.000, 395.00)
PT. 2
(0.100, 380.50)
PT. 3
(0.200, 351.11)
PT. 4
(0.704, 297.83)
PT. 5
(1.587, 211.47)
PT. 6
(3.175, 191.75)
PT. 7
(6.349, 25.60)
PT. 8
(12.698, 111.73)

PRESS F FOR APPROX. CURVE

Fig. 5.17a
Display of Linear Search
within the FR Method.



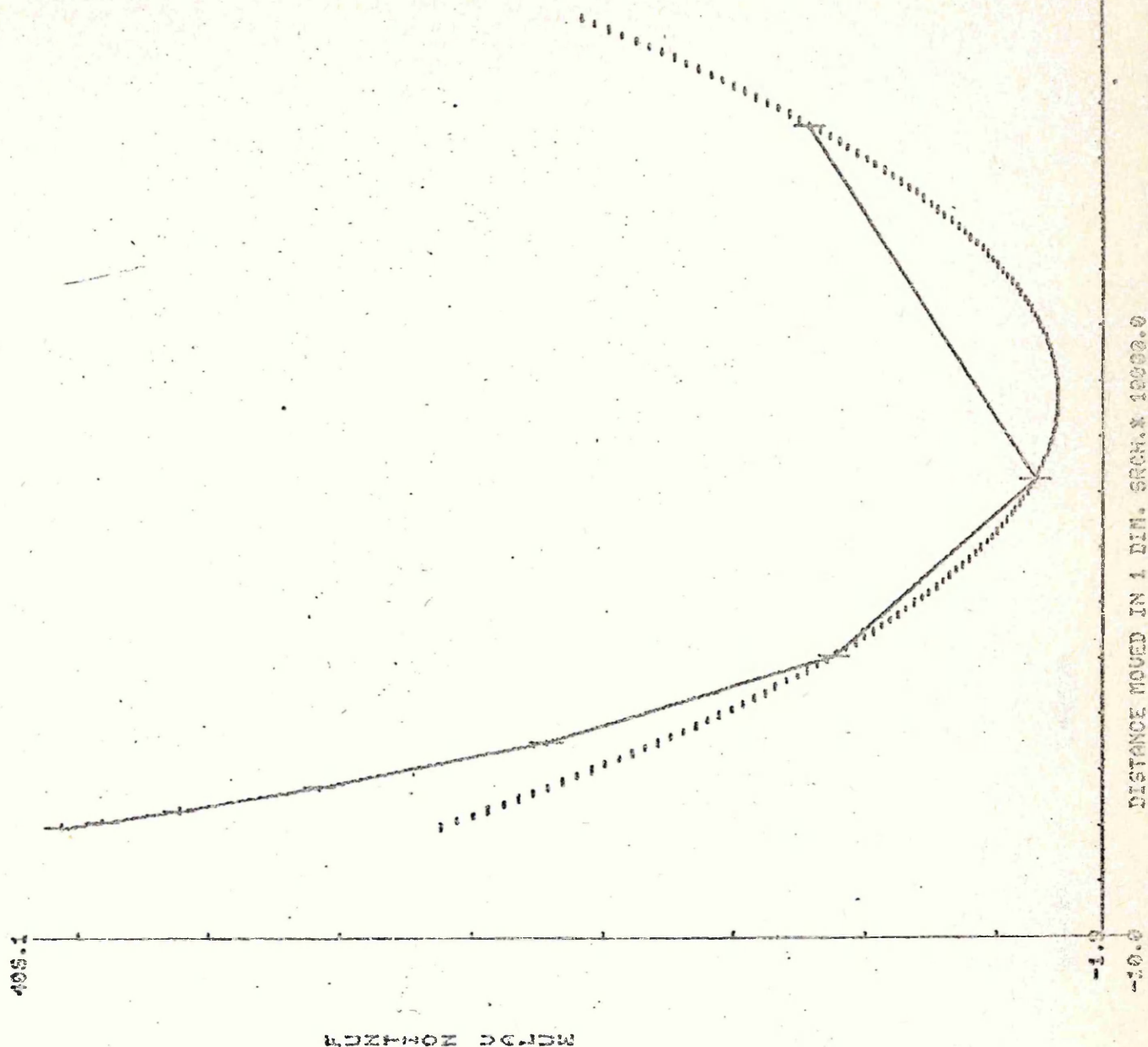
PRESS B FOR CURSE
PRESS C TO CNT. WITH CRSR. INPUT
PRESS D TO CONTINUE

PT. 1
(0.029, 395.06)
PT. 2
(0.193, 390.58)
PT. 3
(0.397, 351.11)
PT. 4
(0.794, 297.83)
PT. 5
(1.587, 211.47)
PT. 6
(3.175, 101.75)
PT. 7
(6.349, 25.60)
PT. 8
(12.698, 11.73)

PRESS F FOR APPROX. CURVE
FUNCT. VAL. AFTER CUB. SEARCH=
F= 20.69
MIN DIS= 0.00035523
PRESS D TO CONTINUE

CONVERGED AS FOLLOWS
NO. OF ITERATION= 33
FMIN.= 0.00000
VALUE OF VAR. AT MIN.
X= 0.9993 0.9990

Fig. 5.17b
Display of Linear Search and
Approximate Curve of Rosenbrock
Function.



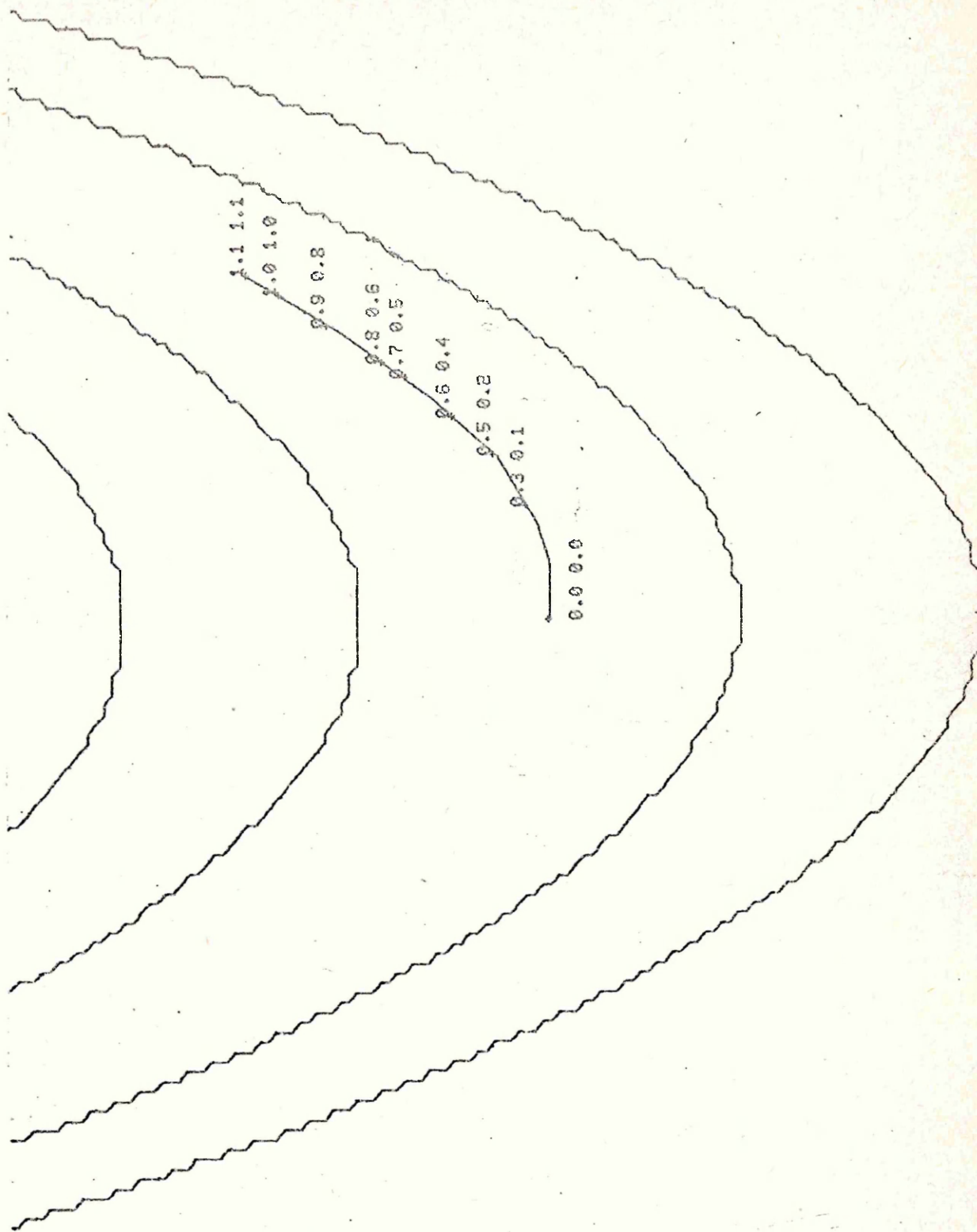
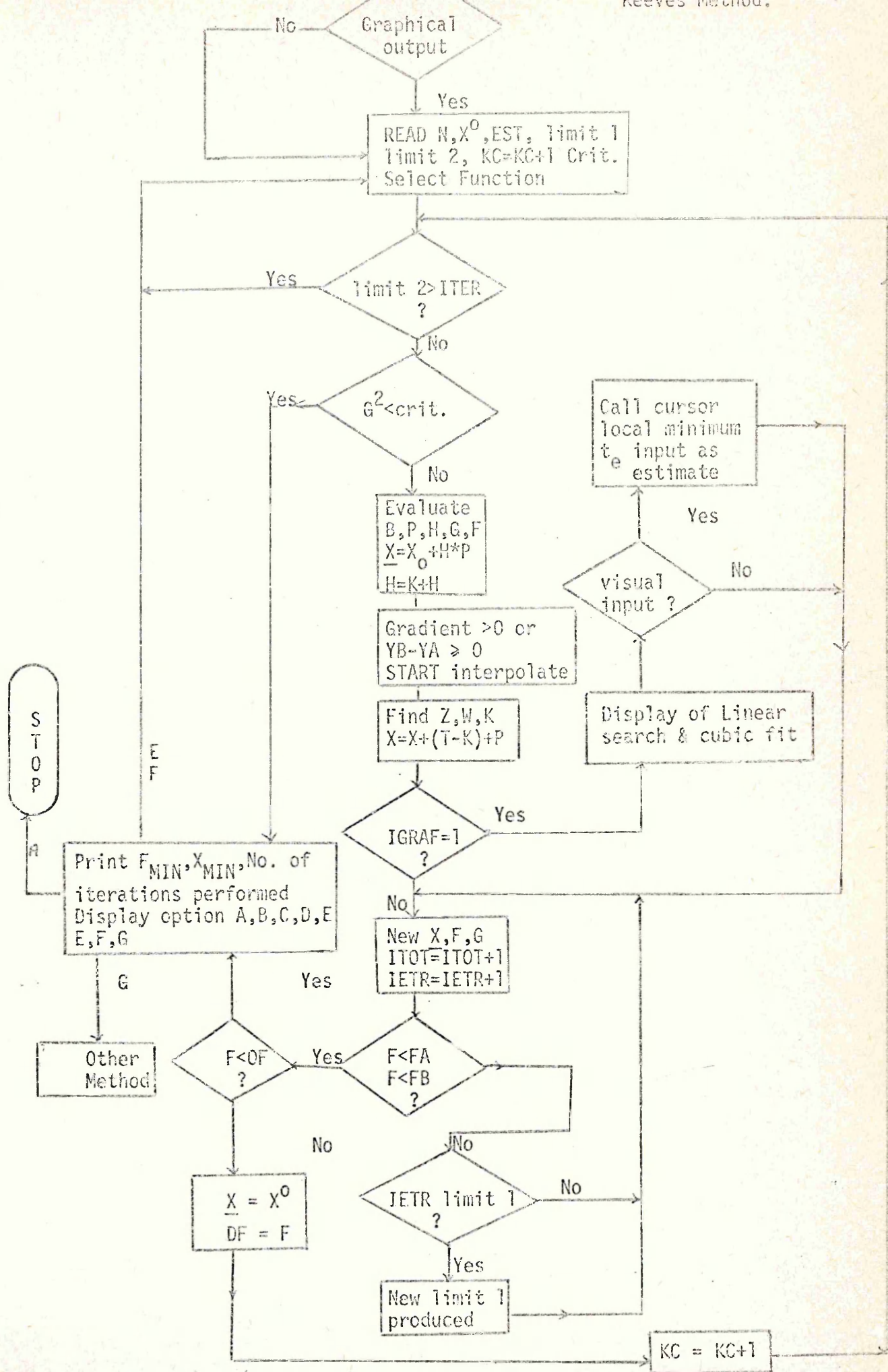


Fig. 5.18 Contours and Minimization Path of Rosenbrock Function.



The direction of movement is that from the vertex with the maximum value of function through a point half way between the two remaining vertices. The distance moved is such that, initially, the new search position also forms an equilateral triangle with the common base. The sequence is repeated, with automatically generated contraction and expansion of the simplex also being undertaken when appropriate.

This method is applicable to all continuous functions but it requires a large number of function evaluations, that is (2^{N+1}) function evaluations for each iteration.

The subroutines for this method are those provided by NAG library. There are three subroutines, two of which are supplied by the user. The first one is subroutine 'FUNCT' which calculates the value of the function for any set values of the variables. The second one is the subroutine 'MONIT' which is involved with the printing out of the results of searches. This subroutine has been modified to provide the user with a graphical presentation of the search path in any two nominated dimensions.

To activate the process the initial values of the variables are selected, and the user must specify the number of variables, maximum number of iterations and the axial variables, i.e. to specify the plane for presenting the search path.

The search pattern can either be presented in complete form, or the user may wish to control the build-up of the pattern if hard copies of each individual step are required. The user decides by pressing the appropriate key after the first triangle has been presented. Other facilities such as change of initial points, change of window, change of method etc., are also included in this program,

and these are presented in the form of menu of options, when the optimization procedure is terminated.

Figure 5.20 shows the flow chart of this program. This includes the basic idea of the simplex method.

Figures 5.21 (a), 5.21 (b) and 5.21 (c) show three steps of the search pattern for the Rosenbrock function being built up in individual steps. Figure 5.21 (d) shows the complete search pattern for the same function with the starting points of

$$x_1 = 1.1 \quad , \quad x_2 = -1.0.$$

Figure 5.22 shows the Rosenbrock function with a different starting point

$$x_1 = 0.2 \quad , \quad x_2 = 0.2.$$

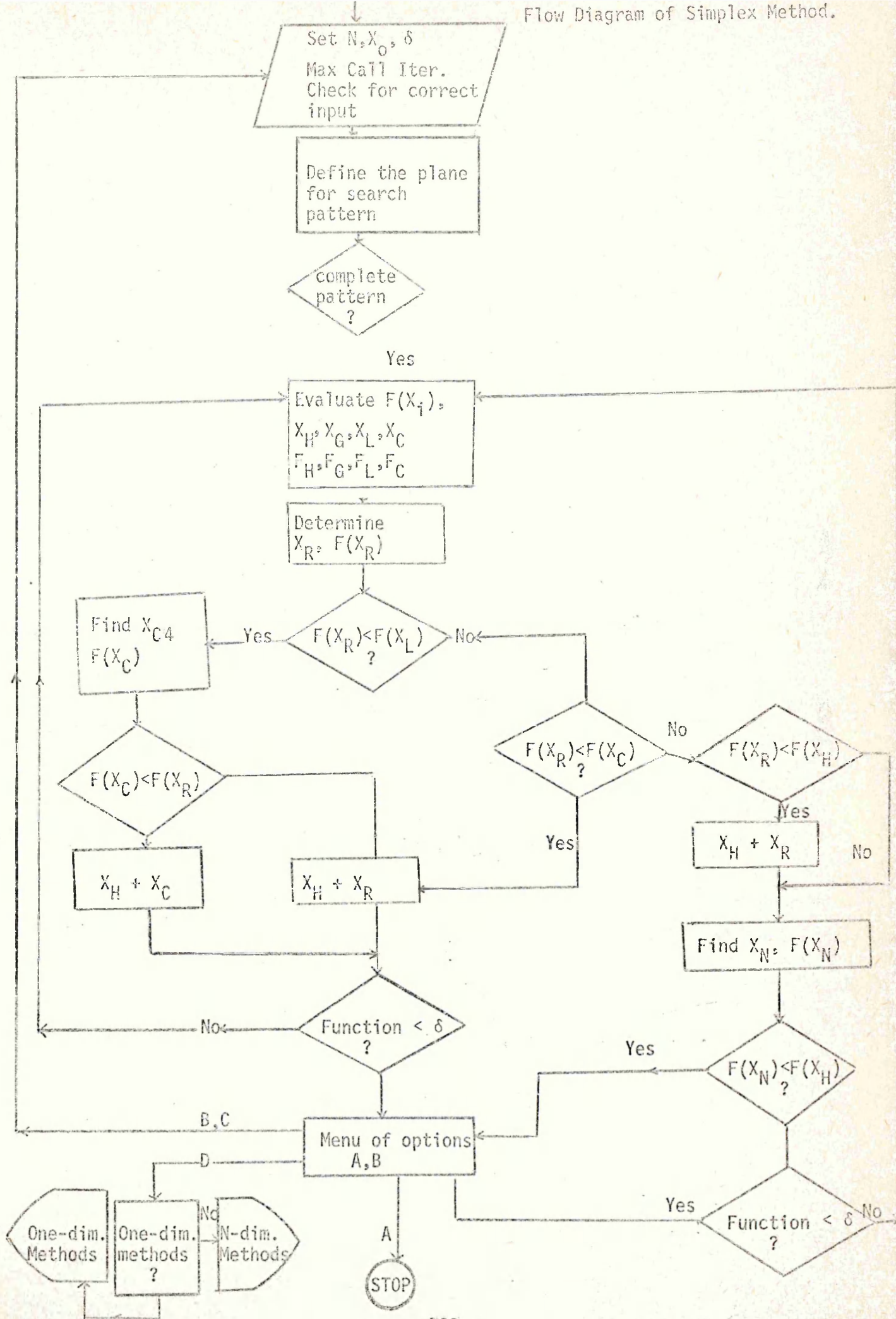
It can be seen that although the starting values are very different the calculated minimum values of x_1 and x_2 are nearly the same.

5.4 Plane Analysis Method

This method involves drawing contours of an N-dimensional function on different two-dimensional planes and then by visual analysis of these planes and contours a minimum is obtained.

The program for this method utilises the contour plotting routine described earlier.

The minimum of an N-dimensional function ($N > 2$), is obtained by successive direct searches on and between defined planes. The approach is applicable to general functions, but as is customary, it will be described and developed theoretically with respect to quadratic form in N-dimensions.



Rosenbrock Function

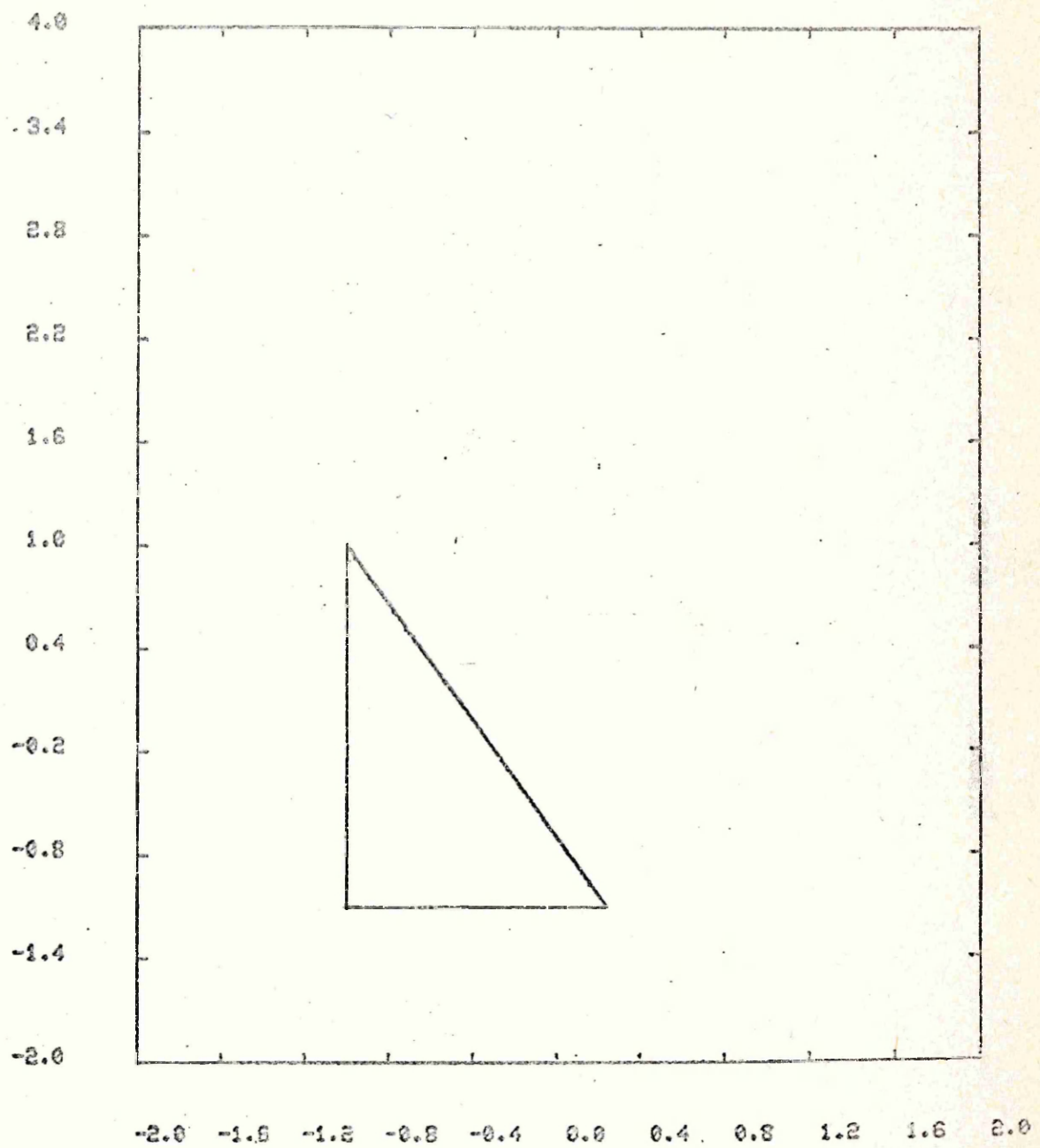


Fig. 5.21a First Step of Simplex Method Search Pattern.

Rosenbrock Function

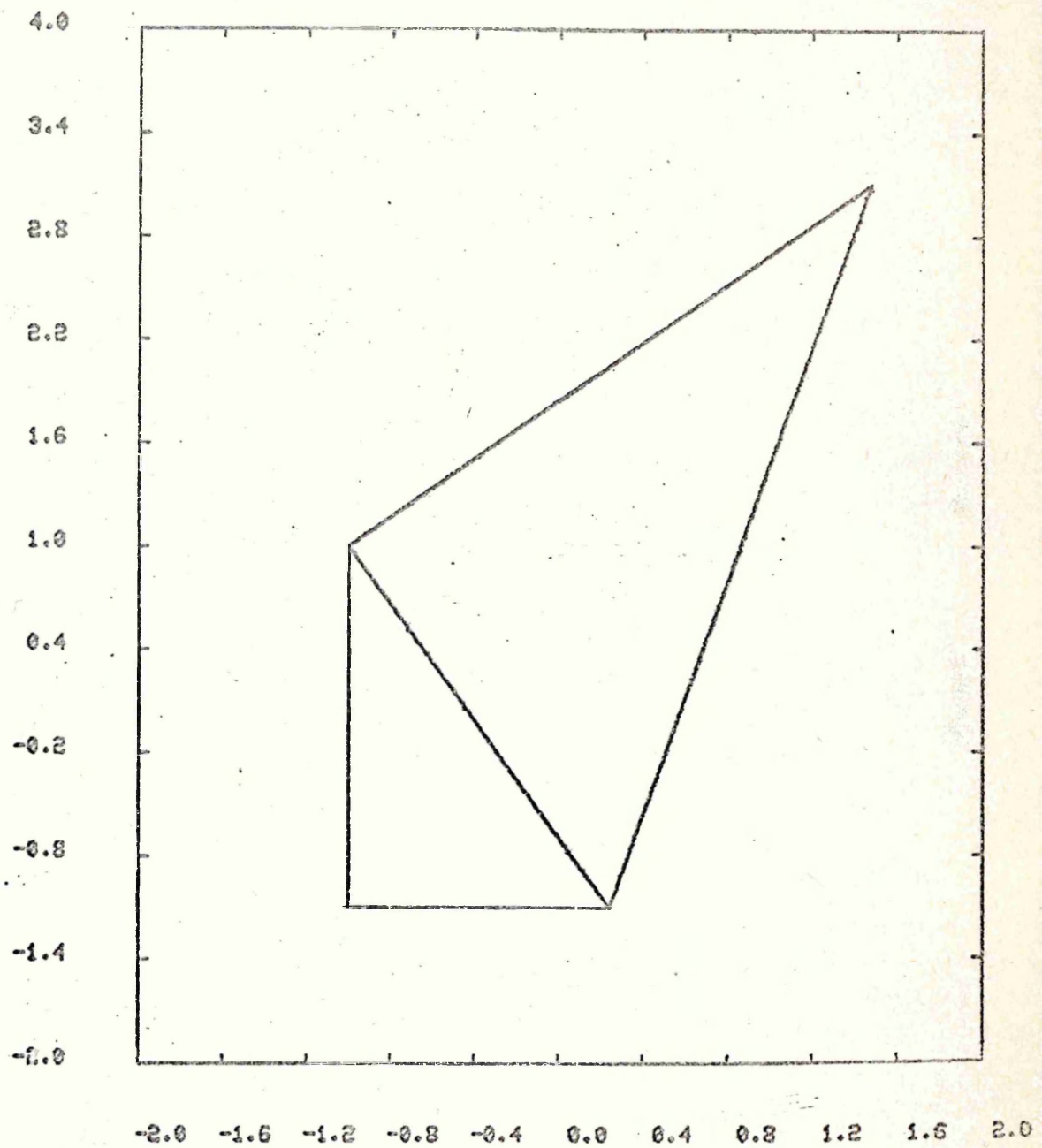


Fig. 5.21b Second Step of Simplex Method Search Pattern.

Rosenbrock Function

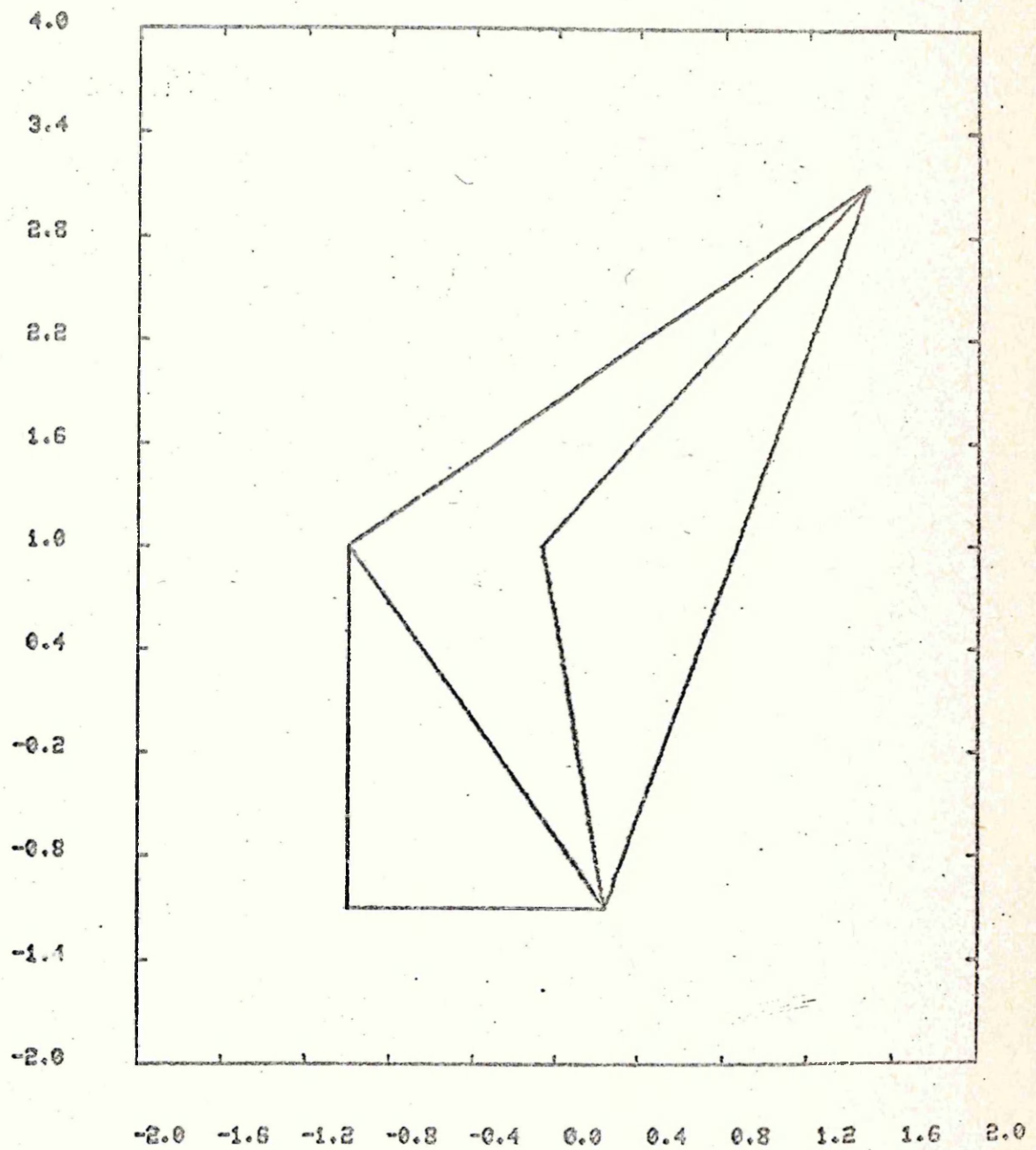
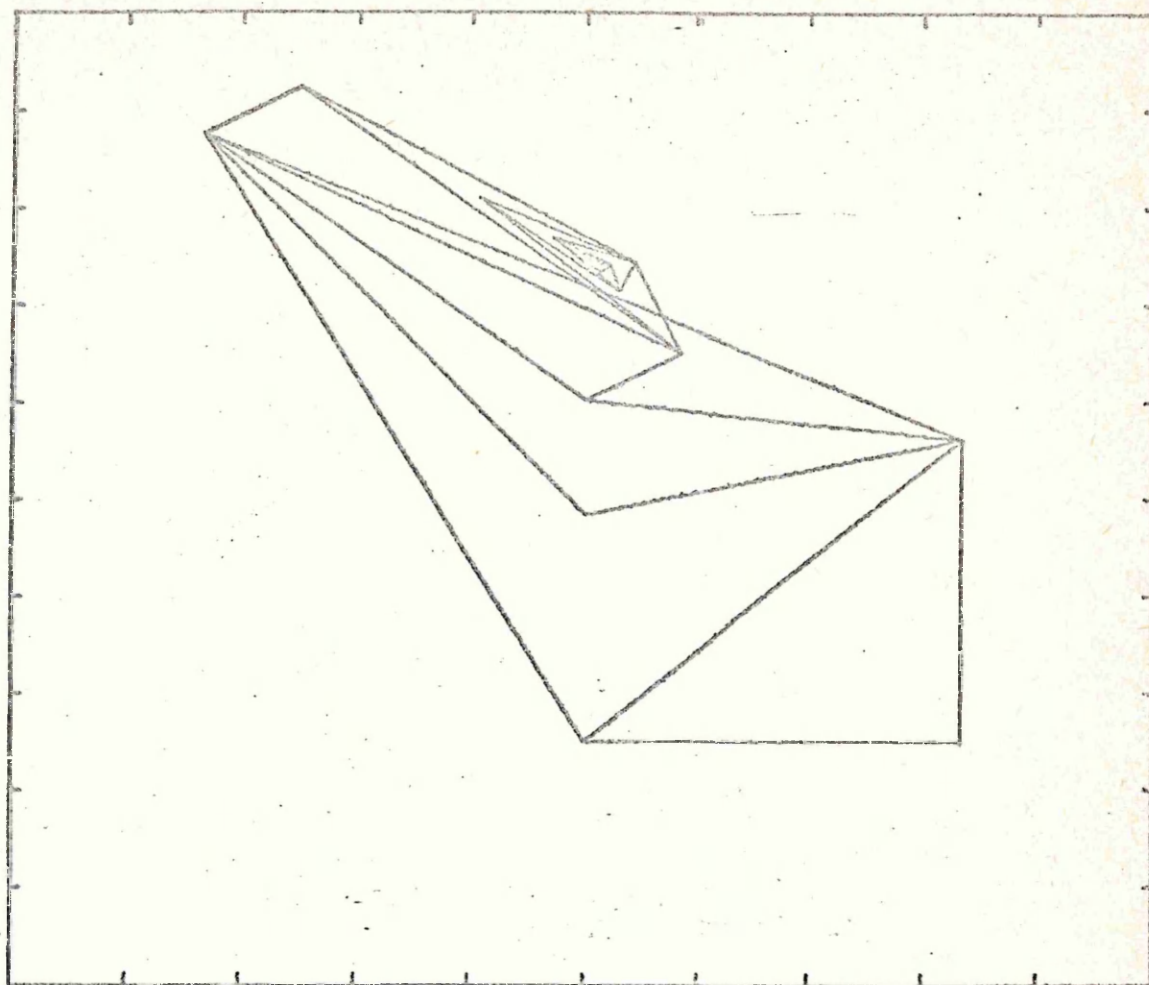


Fig. 5.21c Third Step of Simplex Method Search Pattern.

FUNCTION VALUE IS 0.0000
AT POINT 0.9999 0.9999

4.0
3.4
2.8
2.2
1.6
1.0
0.4
-0.2
-0.8
-1.4
-2.0

Fig. 5.21d
A Complete Search Pattern of
Rosenbrock Function using
Simplex Method.



-2.0 -1.0 -1.2 -0.8 -0.4 0.0 0.4 0.8 1.2 1.6 2.0

FUNCTION VALUE IS 0.0000
AT POINT 1.0001 1.0002

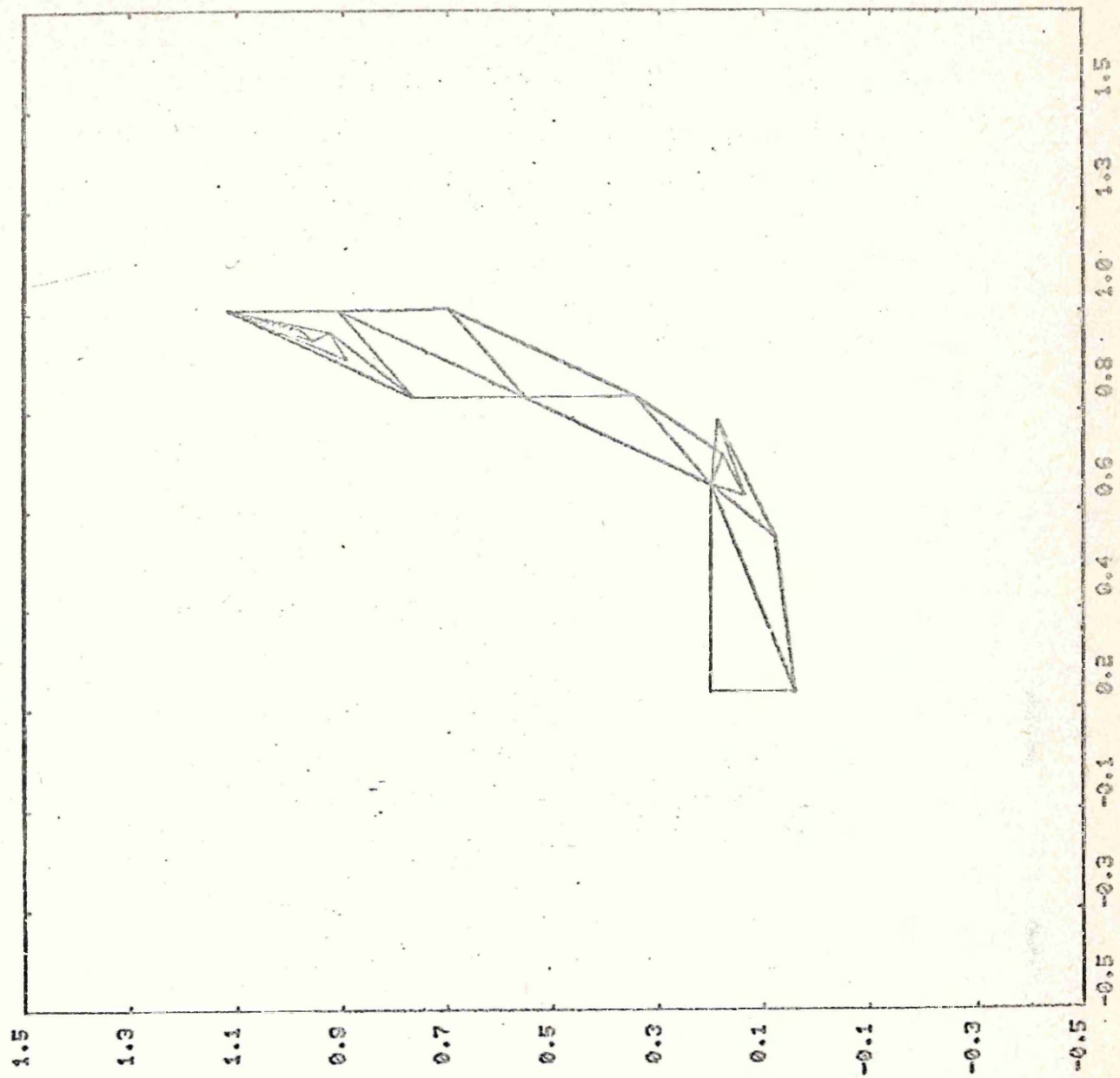


Fig. 5.22
Search Pattern of Rosenbrock
Function with Different
Starting Points.

5.4.1 Development for quadratic forms

The minimum of a quadratic function of three variables for example is found by setting one of the variables at two fixed values and locating the minimum of the two slices of the plane. A search along the line joining the two set of points will find the true minimum, i.e., when $N = 3$ and the plane is $x_1 x_2$ then let

$$x_3 = P_1, \text{ the minimum is } A(a_1, a_2, a_3)$$

$$x_3 = P_2, \quad " \quad " \quad " \quad B(b_1, b_2, b_3)$$

A search along the line AB will give the minimum $C(c_1, c_2, c_3)$.

Therefore to find the minimum two plane slices and one search are required.

Consider, now a function in four variables. Let

$$x_4 = G_1$$

$$x_3 = P_1, \text{ the first minimum is at } A(a_1, a_2, a_3, a_4)$$

$$x_3 = P_2, \text{ the second minimum is at } B(b_1, b_2, b_3, b_4).$$

A one-dimensional search between A and B gives $C(c_1, c_2, c_3, c_4)$.

Now let $x_4 = G_2$

$$x_3 = P_3, \text{ the first minimum is at } A'(a_1', a_2', a_3', a_4')$$

$$x_3 = P_4, \text{ the second minimum is at } B'(b_1', b_2', b_3', b_4')$$

One-dimensional search between A' and B' gives the new minimum

$C'(c_1', c_2', c_3', c_4')$. Now a third search between C and C' gives

$D(D_1, D_2, D_3, D_4)$ the global minimum, i.e. four plane slices and three searches are required.

Therefore it can be concluded that for a N-dimensional quadratic function, 2^{N-2} plane slices and N-1 linear searches are required.

The mathematical basis for the above procedure for a quadratic function is as follows:

Theorem

The minimum of a quadratic function F in $(n+1)$ variables can be found by searching along a line joining two points, each obtained by setting one of the variables to an arbitrary constant and minimizing the resultant form in n variables.

The minimum of F along the line corresponds exactly to the global minimum of the quadratic form in $(n+1)$ variables.

Proof

Let $F = C + \underline{b}' \underline{x} + \frac{1}{2} \underline{x}' \underline{A} \underline{x}$ be a quadratic form in $(n+1)$ variables. Suppose \underline{b} and \underline{A} are partitioned as follows: (1)

$$\underline{b} = \begin{bmatrix} \underline{b} \\ \underline{b} \end{bmatrix}, \quad \underline{A} = \begin{bmatrix} A & \underline{a} \\ \underline{a}' & a \end{bmatrix} \quad (2)$$

so that \underline{a} and \underline{b} have dimension $(n \times 1)$, A is $(n \times n)$, and a and b are scalars.

Now the minimum of F occurs at

$$\underline{x}^* = -\underline{A}^{-1} \underline{b} \quad (3)$$

and since it can be shown that,

$$\underline{A}^{-1} = \begin{bmatrix} A^{-1} + \frac{1}{D} A^{-1} \underline{a} \underline{a}' A^{-1} & -\frac{1}{D} A^{-1} \underline{a} \\ -\frac{1}{D} \underline{a}' A^{-1} & \frac{1}{D} \end{bmatrix} \quad (4)$$

where $D = a - \underline{a}' A^{-1} \underline{a}$,

then $\underline{\bar{x}}^*$ may be written in partitioned form as,

$$\underline{\bar{x}}^* = \begin{bmatrix} \underline{x}^* \\ \underline{x} \end{bmatrix} = \begin{bmatrix} -A^{-1} \underline{b} - \frac{1}{D} A^{-1} \underline{a} \underline{a}' A^{-1} \underline{b} + \frac{b}{D} A^{-1} \underline{a} \\ \frac{1}{D} \underline{a}' A^{-1} \underline{b} - \frac{b}{D} \end{bmatrix} \quad (5)$$

where $\underline{\bar{x}}^*$ has dimension $(n \times 1)$ and x is (1×1) .

However, if one of the original variables x is set equal to an arbitrary constant K , and the resultant quadratic form in n variables is minimized, this gives rise to a point (in the original $n+1$) dimensional space).

$$\begin{bmatrix} \underline{y}^* \\ K \end{bmatrix} = \begin{bmatrix} -A^{-1} \underline{b} - K A^{-1} \underline{a} \\ K \end{bmatrix} \quad (6)$$

Any point on the line joining two such points (corresponding to arbitrary constants K_1 and K_2) may be written,

$$\underline{\bar{y}} = \begin{bmatrix} \underline{y}_1^* \\ K_1 \end{bmatrix} + \lambda \begin{bmatrix} \underline{y}_2^* - \underline{y}_1^* \\ K_2 - K_1 \end{bmatrix} = \begin{bmatrix} -A^{-1} \underline{b} - A^{-1} \underline{a} (K_1 + \lambda (K_2 - K_1)) \\ K_1 + \lambda (K_2 - K_1) \end{bmatrix} \quad (7)$$

where $0 < \lambda \leq 1$ and such that $\lambda = 0$ gives $\underline{\bar{y}} = \begin{bmatrix} \underline{y}_1^* \\ K_1 \end{bmatrix}$, and

$\lambda = 1$ gives $\underline{\bar{y}} = \begin{bmatrix} \underline{y}_2^* \\ K_2 \end{bmatrix}$.

Note that $\frac{\lambda}{1} = \frac{x - K_1}{K_2 - K_1}$ however,

$\therefore x = K_1 + \lambda(K_2 - K_1)$, enabling equation (7) to be written as

$$\bar{y} = \left[\frac{-A^{-1}\underline{b} - x A^{-1}\underline{a}}{x} \right] \quad (8)$$

i.e., the minimum of F with respect to λ will occur simultaneously with the minimum with respect to x .

Substituting (8) into (1) gives F as a function of the single variable x ,

$$\begin{aligned} F &= C + \left[\frac{\underline{b}}{\underline{b}} \right] \bar{y} + \frac{1}{2} \bar{y}' A \bar{y} \\ &= C - \underline{b}' A^{-1} \underline{b} - x \underline{b}' A^{-1} \underline{a} + bx + \frac{1}{2} \underline{b}' A^{-1} \underline{b} + x \underline{b}' A^{-1} \underline{a} - x \underline{a}' A^{-1} \underline{b} \\ &\quad - x^2 \underline{a}' A^{-1} \underline{a} + x^2 \underline{a} \\ &= C - \underline{b}' A^{-1} \underline{b} - x \underline{b}' A^{-1} \underline{a} + bx + \frac{1}{2} \underline{b}' A^{-1} \underline{b} - x^2 \underline{a}' A^{-1} \underline{a} + x^2 \underline{a} \end{aligned}$$

on account of A^{-1} being symmetric.

At the minimum of F with respect to x ,

$$0 = -\underline{b}' A^{-1} \underline{a} + b + x - \underline{a}' A^{-1} \underline{a} + \underline{a} = -\underline{b}' A^{-1} \underline{a} + b + x D$$

therefore

$$x^* = \frac{1}{D} (\underline{b}' A^{-1} \underline{a} - b) \quad (9)$$

Hence,

$$\bar{y}^* = \left[\frac{-A^{-1}\underline{b} - \frac{1}{D} A^{-1}\underline{a} \underline{b}' A^{-1}\underline{a} + \frac{b}{D} A^{-1}\underline{a}}{\frac{1}{D} \underline{b}' A^{-1} \underline{a} - \frac{b}{D}} \right] \quad (10)$$

and, remembering that A^{-1} is symmetric, this corresponds exactly with the location of the true $(n+1)$ dimensional minimum given by \bar{x}^* in equation (8). It is, of course, independent of K_1 and K_2 .

Finally, note that in practical terms, confirmation of the minimum can be obtained at $x = x^*$ by minimizing the n-dimensional quadratic form resulting from setting $K = x^*$. It can be seen from equation (6) that this also gives the point \bar{x}^* .

5.4.2. A function of quadratic form

The function which has been minimized using the theory for a quadratic form is;

$$F = (4x_1 + 3x_2 + 2x_3 + x_4 - 20)^2 + (-x_1 + x_2 - x_3 + x_4 - 2)^2 \\ + (x_1 + 2x_2 - x_3 - x_4 + 2)^2 + (2x_1 - x_2 + 2x_3 - x_4 - 2)^2$$

Figures 5.23(a) and 5.23(b) show the contours of the function on two different planes, and the minimum on each plane is located. The variable x_4 is set at a fixed value, $x_4 = 0$, and two values are given to x_3 , i.e. $x_3 = 0$ and $x_3 = 1$. The co-ordinates of the two minimum points are used to find the minimum along the line joining the two points. Figure 5.23(c) shows the curve of the linear search and the function value at $F = 5.36$.

Figures 5.24(a) and 5.24(b) show second set of contours on the planes of $x_4 = 1$ with $x_3 = 0$ and $x_3 = 1$. The two minimum points are located visually, and a linear search is performed as before to find the minimum along the line joining the two points. Figure 5.24(c) shows the curve, the function value and the co-ordinates of the minimum.

The last stage of the minimization process is to find the minimum between the two sets of values which is also the minimum value of the function. Figure 5.25 shows the curve, the values of the variables and the function minimum.

PAGE 3 FOR F.P. METHOD
PRESS I FOR 1,2,3RD CURVES

PRESS I FOR NEW PAGE & WINDO
PRESS J FOR NEW PAGE & COOR.

WITH CURSOR A...I, AS BEFORE

RANGE OF X FORMAT F7.3
2.525 9.000 -9.000 2.625
RANGE OF Y FORMAT F6.3
1.95125 9.000 -9.000 1.925
NON AXIAL DIMS, F7.3
DIM 3
0.000
DIM 4
0.000

PT 1- 2.625 1.925 0.0 0.0 94.747
XPT 1- 2.733 1.750 0.0 0.0 94.194

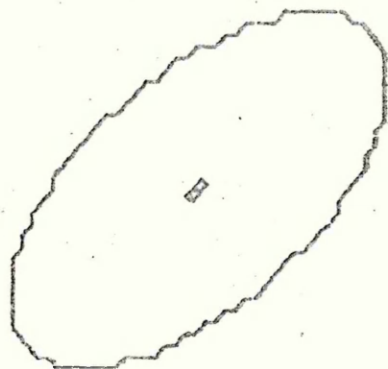


Fig. 5.23a
Contour of Function 5.1
on the (x_1, x_2) plane

$x_3 = 0.0$
 $x_4 = 0.0$

FMIN= 412.000
FNC.EVAL. KN1= 1452
FMIN= 94.747
FNC.EVAL. KN1= 1576

PRESS I FOR CURSOR
PRESS C FOR F.P. METHOD
PRESS E FOR 1,2,3RD CURVES

PRESS F,G,H FOR 2,3,4TH XN INPUT
PRESS I FOR NEW PAGE & WINDO
PRESS J FOR NEW PAGE & COOR.

WITH CURSOR,A...I,AS BEFORE

RANGE OF X FORMAT F6.3
2.378 9.000-9.000 2.378
RANGE OF Y FORMAT F6.3
1.283 9.000-9.000 1.283
NON AXIAL DIMS.F6.3
DIM 3
1.000
DIM 4
0.000

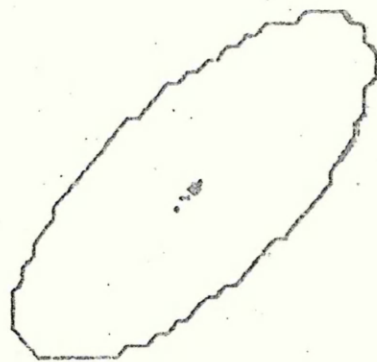
PT 1- 2.125 1.633 1.000 0.0 114.535
F NOT REDUCED F= 116.742
PT 2- 2.125 1.453 1.000 0.0 113.900
PT 4- 2.378 1.283 1.000 0.0 113.283
F NOT REDUCED F= 113.456

Fig. 5.23b
Contour of Function
5.1 on the
(x_1, x_2) plane

$x_3 = 1.0$

$x_4 = 0.0$

FMIN= 334.000
FNC.EVAL. KN1= 374
FMIN= 113.282
FNC.EVAL. KN1= 490



ENTER L TO DRAW NEW CURVE & WINDOW
PRESS F TO DRAW NEW CONTOUR, SAME WINDOW
PRESS G FOR ANEW WINDOW

ENTER L FOR CURSOR
PRESS B TO CONT. GRAPH NEW WZ
PRESS C TO DRAW NEW GRAPH & WINDOW
PRESS D TO DRAW NEW GRAPH & WINDOW

GRAPH OF FUN. V DIST.

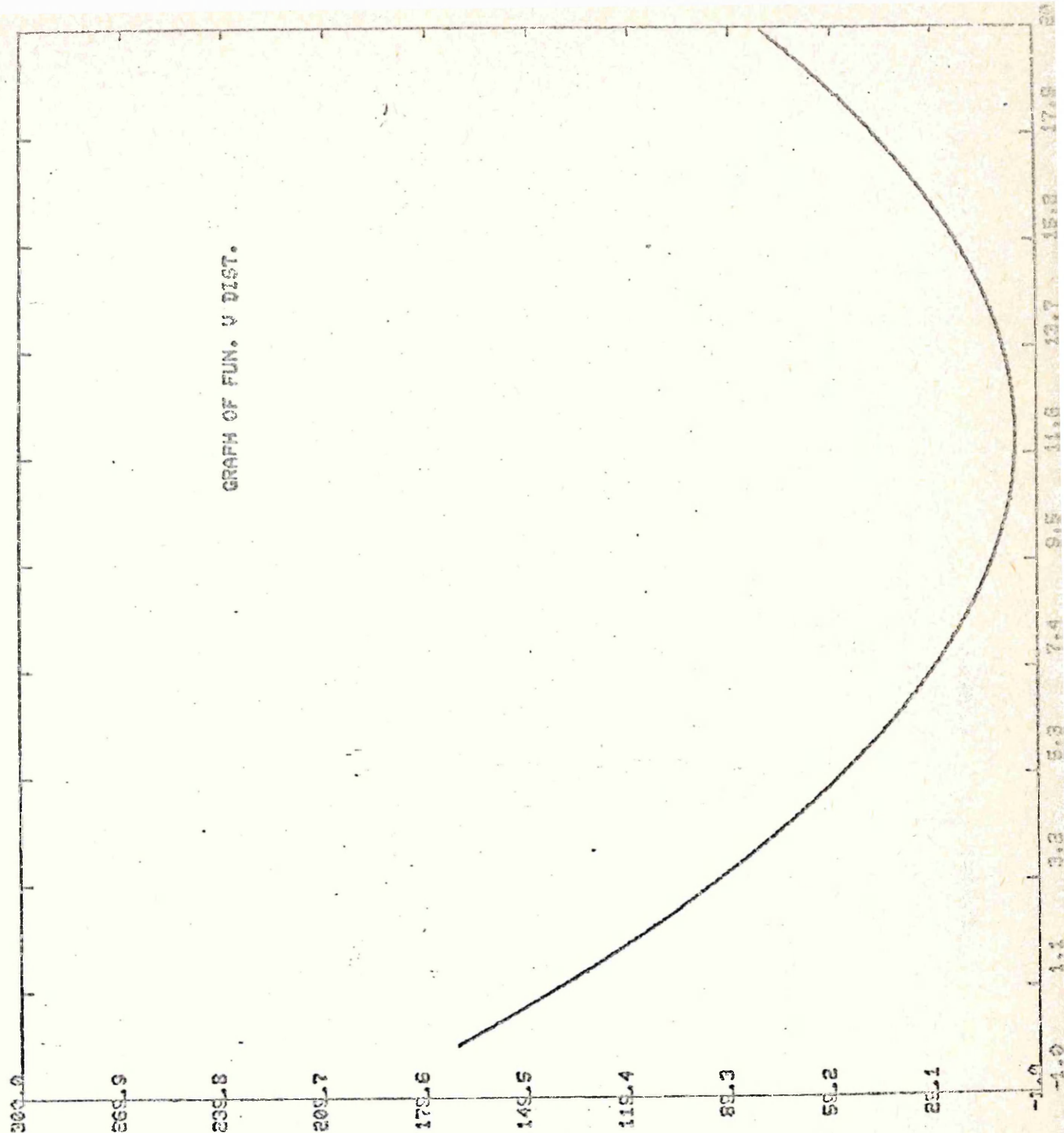


Fig. 5.23c

Direct Search between
the Minima of the Two
Planes with $x_4 = 0.0$.

RESULT OF PROBLEM
MIN. VAL. OF x_1, x_2, \dots
-7.678 7.673 11.970 9.9
LAMDA=11.97 F= 5.3619

PRESS ANY OTHER KEY FOR START
WITH CURSOR, A...I, AS BEFORE

ENTER CURSOR FINDING
PRESS F, G, H FOR 2, 3, 4TH XN INPUT
PRESS I FOR NEW PAGE & WINDO
PRESS J FOR NEW PAGE & COOR.

PRESS R FOR CURSOR
PRESS C FOR F.P. METHOD
PRESS D FOR 1, 2, 3RD CURVES

RANGE OF X FORMAT F6.3
3.492 9.000 9.000 3.492
RANGE OF Y FORMAT F6.3
0.457 9.000 9.000 0.467
NON AXIAL DIMS, F6.3
DIM 3
0.000
DIM 4
1.000

PT 1= 3.239 0.817 0.0 1.000 80.287
PT 2= 3.422 0.467 0.0 1.000 79.583
F NOT REDUCED F= 80.083
PT 1= 3.340 0.583 0.0 1.000 79.283

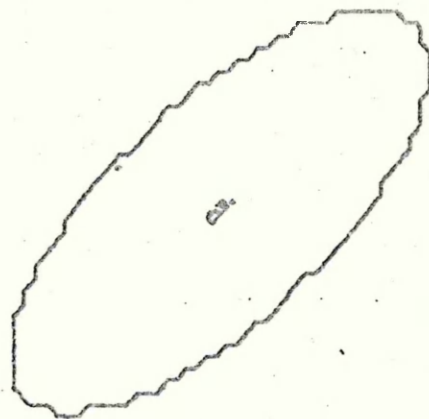


Fig. 5.24a

Contour of Function
5.1 on the (x_1, x_2)
plane

$x_3 = 0.0$

$x_4 = 1.0$

FMIN= 372.000
FNC.EVAL. KNI= 410
FMIN= 79.587
FNC.EVAL. KNI= 528

PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C FOR F.P. METHOD
 PRESS D FOR 1,2,3RD CURVES
 PRESS E FOR MIN.COOR FINDING
 PRESS F,G,H FOR 2,3,4TH XN INPUT
 PRESS I FOR NEW PAGE & WINDO
 PRESS J FOR NEW PAGE & COOR.
 WITH CURSOR,A...I,AS BEFORE
 PRESS ANY OTHER KEY FOR START

RANGE OF X FORMAT FG.3
 0.2.323 9.000-9.000 2.323
 RANGE OF Y FORMAT FG.3
 1.400 9.000-9.000 1.400
 NON AXIAL DIMS.FG.3
 DIM 3
 1.000
 DIM 4
 1.000

PT 1= 2.323 1.400 1.000 1.000 76.361
 PT 1= 2.429 1.108 1.000 1.000 75.479
 F NOT REDUCED F= 75.523

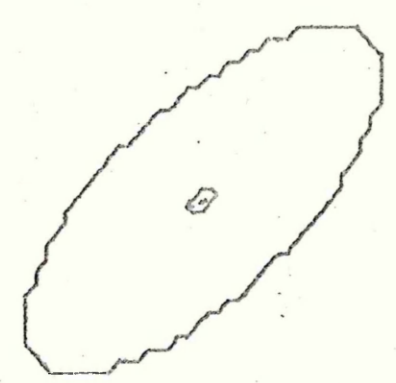


Fig. 5.24b.
 Contour of Function 5.1
 on the (x_1, x_2) plane

$x_3 = 1.0$
 $x_4 = 1.0$

FMIN= 294.000
 FNC.EVAL. KN1= 898
 FMIN= 76.362
 FNC.EVAL. KN1= 1026

PRESS F TO DRAW NEW CONTOUR, SAME WINDOW
PRESS G FOR ANEW WINDOW

PRESS F FOR CURSOR
PRESS C TO CONT. GRAPH NEW MI
PRESS D TO DRAW NEW GRAPH & WINDOW

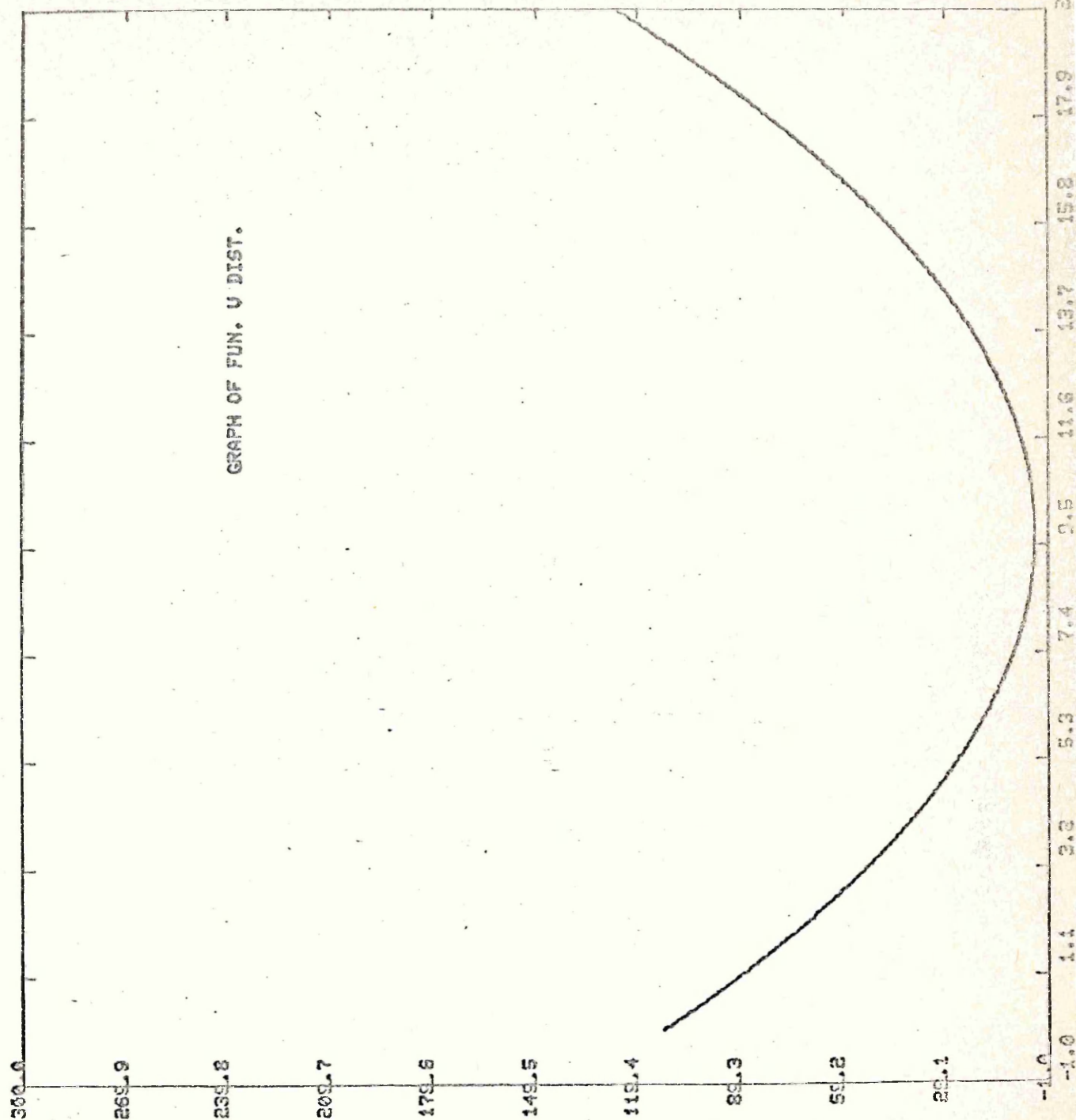


Fig. 5.24c

RESULT OF PROBLEM
MIN. VAL. OF X_1, X_2
-5.037 5.239 9.700 1.000
LAMBDA= 9.70 F= 2.8875

Direct Search between
the minima of the
two planes with
 $x_4 = 1.0$.

PRESS F TO DRAW NEW CONTOUR, SAME WINDOW
PRESS G FOR ANEW WINDOW

PRESS C TO CONT. GRAPH NEW HT
PRESS D TO DRAW NEW GRAPH & WINDOW

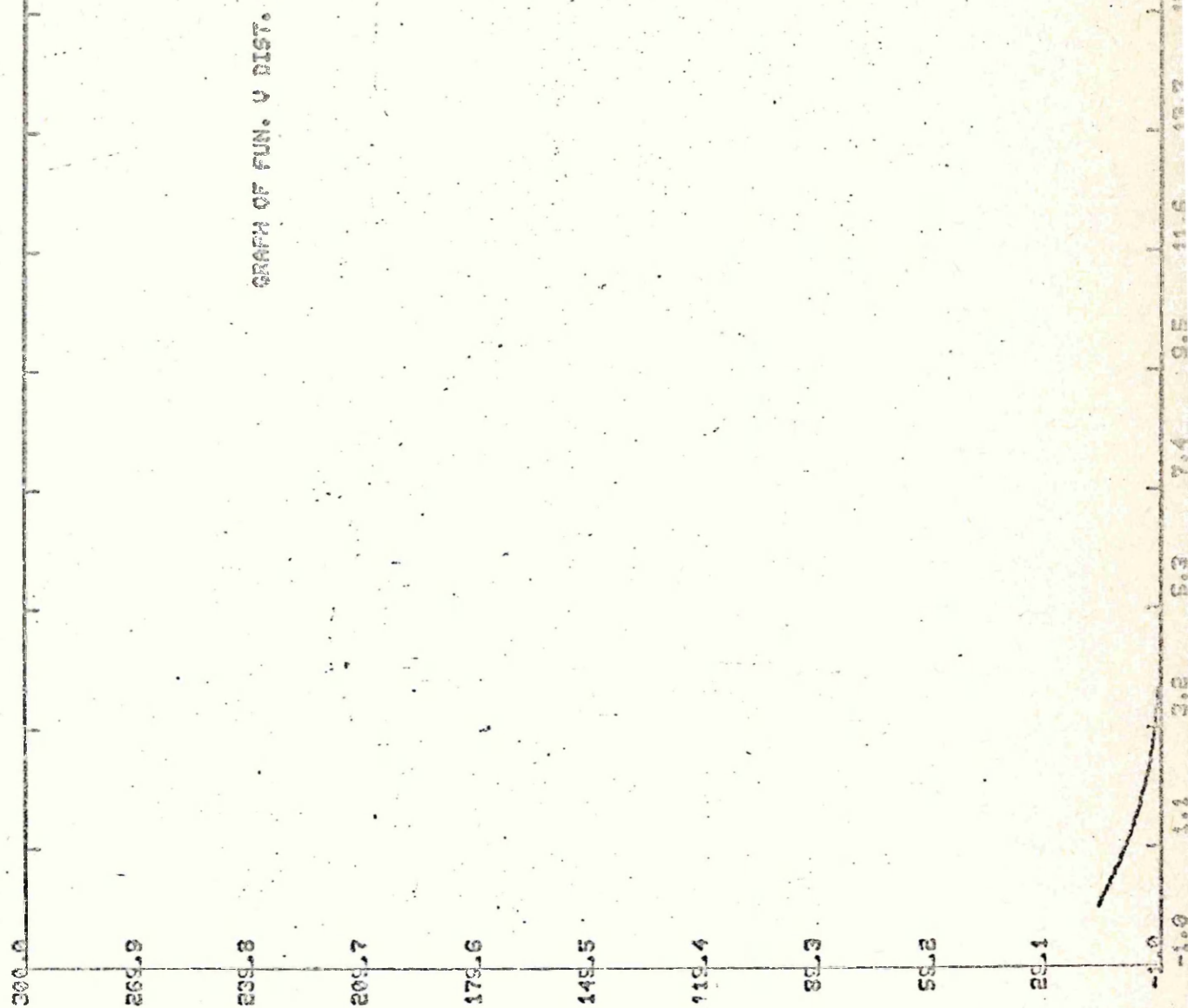


Fig. 5.25

Final Direct Search to
Obtain the Minima.

RESULT OF PROBLEM
MIN. VAL. OF X_1, X_2
1.000 2.000 3.000 4.000
LAMBDA = 4.000 ∇ 0.0000

5.4.3 Application to functions of non-quadratic form

The minimum value of functions which do not possess quadratic forms and constrained functions can be found using a modification of the approach previously described.

Consider a general three-dimensional function. To perform a direct search two sets of values are required, that is one set of values from each plane.

To start the process the user specifies the plane through which the minimization process is carried out, and the ranges of planar variables. The other variable, referred to as the non-planar variable, is set to a fixed value, and the contours of the function for this set of values is plotted. The minimum value of the function for this plane is located. Now a new value is assigned to the non-planar variable and the second minimum value is obtained. With these two sets of values the user can now proceed with the linear search.

The linear search process is the same as the one used for quadratic form functions.

To start the process the user must define the range of the window, the step size for the search and the number of iterations. The step size and the number of iterations can also be set automatically in the program if instructed by the user. The iteration number allows the user to find out how the search is proceeding. When the number of iterations reaches the limit, the user can either reduce the step size, input a new iteration limit, and proceed with the search or ask the menu of options to be displayed.

The set of values found by the linear search is used to plot a contour of the function on a new plane and the minimum is obtained

as before. The first set of values is now discarded and a direct search is performed between the last two sets to find a new minimum. This process is repeated until the required minimum has been reached.

The same idea may be used to obtain the minimum of functions with larger number of variables.

This method was originally introduced to deal with unconstrained N-dimensional functions, but it has now been extended and certain facilities have been incorporated in this program which allow contours of constrained functions to be plotted. The constraints are also drawn which specify the feasible region.

In the initialization stage the user defines types and the number of constraints of each type. Three types of constraints have been declared here:-

(1) Single valued constraints,

$$a_i \leq x_i \leq b_i \quad i = 1, \dots, N$$

In most cases these constraints are not drawn.

(2) Linear constraints,

$$a \leq c_1x_1 + c_2x_2 + \dots + c_jx_j \leq b \quad j = 1, \dots, N$$

(3) Non-linear constraints

The constraints are plotted on the same plane as the functions, and the non-planar variables will have the same values assigned to them when plotting the contours of the function equation.

When the constraints are drawn, the user will have a better knowledge of the feasible region and where the minimum on a particular slice may be.

The minimization process is basically the same as for unconstrained functions, except that when a set of points are found, either by using the cursor to locate the minimum value on the plane slices or by means of the direct search routine, a check is automatically carried out to verify if the points satisfy the constraints or not.

When the points are located visually and the constraints are not satisfied, a message is displayed to point out to the user the variables are not within the feasible region, otherwise, as before the value of the function and the variables are displayed.

When the direct search routine is used, the calculated points are tested at each stage and if any of these values do not satisfy the constraints then, the search is terminated, presenting the user with the curve of the direct search, the results at the stage before the process is terminated and a menu of options from which the user can choose the next step.

In chapter six the minimum of a constrained function is found using the process which has already been described.

Figure 5.26 shows the flow diagram of the plane analysis method.

5.4.4 Test functions

The minimum of the Powell's function was found using the alternative approach. The function is;

$$F = - \{ (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^2 + 10(x_1 - x_4)^4 \}$$

This function has a minimum at $\underline{x}^* = (0.0, 0.0, 0.0, 0.0)$.

Figures 5.27a and 5.27b show two contours of this function. X_1 and X_2 were chosen as the planar variables and X_3 and X_4 are set to fixed values, $X_3 = 1.0$, $X_4 = 1.0$ for the first contour and $X_3 = 0.0$, $X_4 = 0.0$ for the second contour. The minimum of each contour was located using the cursor and a direct search was carried out between the two sets of points. Figure 5.27c shows the curve and the results of this search. The values obtained from the direct search were used to plot a new contour, shown in figure 5.27d. The minimum of this contour was located. The values of the first contour were discarded and a direct search was performed between the last two sets of points, shown in figure 5.27e. Figure 5.27f shows the last contour drawn using the results of the previous direct search. The minimum of this contour was visually located. This value was found to be satisfactory and the process was terminated at this point.

The minimum of Wood's function was also found using the above. The function is;

$$F = -\{100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2) + (1 - x_3)^2 + 10.1(x_2 - 1)^2 + 10.1(x_4 - 1)^2 + 19.8(x_2 - x_1)(x_4 - 1)\}.$$

Figure 5.28a shows the contours of this function. The planar variables are again set at X_1 and X_2 and X_3 and X_4 are both set to (0.0). Figure 5.28b shows a close up of the second contour with its minimum value at the fourth point. Figure 5.28c shows the contour of the function with $X_3 = 1.0$ and $x_4 = 1.0$. Figure 5.28d shows the linear search which was carried out between the two set of minimum values. The results of this direct search were used to draw a new contour shown in figure 5.29 which is a close-up of the contour. The minimum of this contour was visually located and it can be seen that

it is close to the true minimum at $X_1 = 1.0$, $X_2 = 1.0$, $X_3 = 1.0$ and $X_4 = 1.0$.

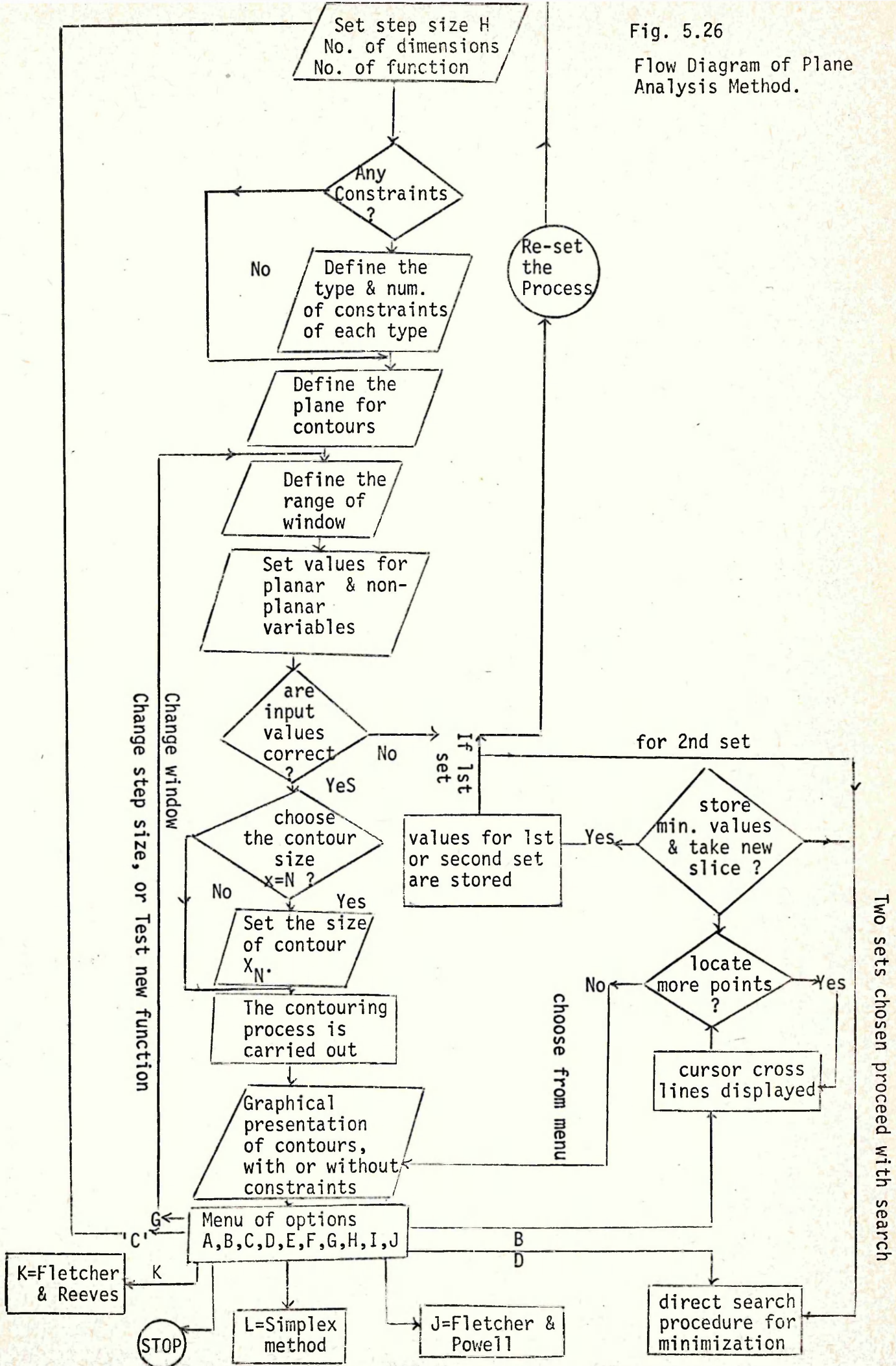
The third function whose minimum was found using this approach was the three-dimensional Rosenbrock function. The function is of the form:-

$$F = -\{100(X_2 - X_1^2)^2 + (1 - X_1)^2 + 100(X_3 - X_2^2) + (1 - X_2)^2\}$$

It has a minimum at $\underline{X}^* = (0.0, 0.0, 0.0)$. Figure 5.30a shows the contour plots of this function of the plane X_1 and X_2 with $X_3 = 0.0$. Figure 5.30b shows the second contour with $X_3 = 1.0$. Figure 5.30c shows the direct search between the minimum values of the contours. The result obtained from the direct search was used to plot a new contour and its minimum was located visually shown in figure 5.30d. The values of the first contour were discarded and a direct search was carried out between the last two set of values as before. The curve and the results of this search are shown in figure 5.30e. The same process was repeated until the minimum was obtained. Figures 5.30f to 5.30m shown the graphical illustration of the process. Figure 5.30m shows the curve of the last direct search with a function value of $F^* = 0.006$. This value was found to be close to the actual minimum of $F = 0.0$. The process was terminated at this stage.

Fig. 5.26

Flow Diagram of Plane Analysis Method.



PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C FOR F.P.METHOD
 PRESS D FOR 1,2,3RD CURVES

PRESS E FOR FUN. VAL. OUTPUT
 PRESS F FOR MIN. COOR. FINDING
 PRESS G,H,I FOR 2,3,4TH XN INPUT
 PRESS K FOR X-Y OUTPUT

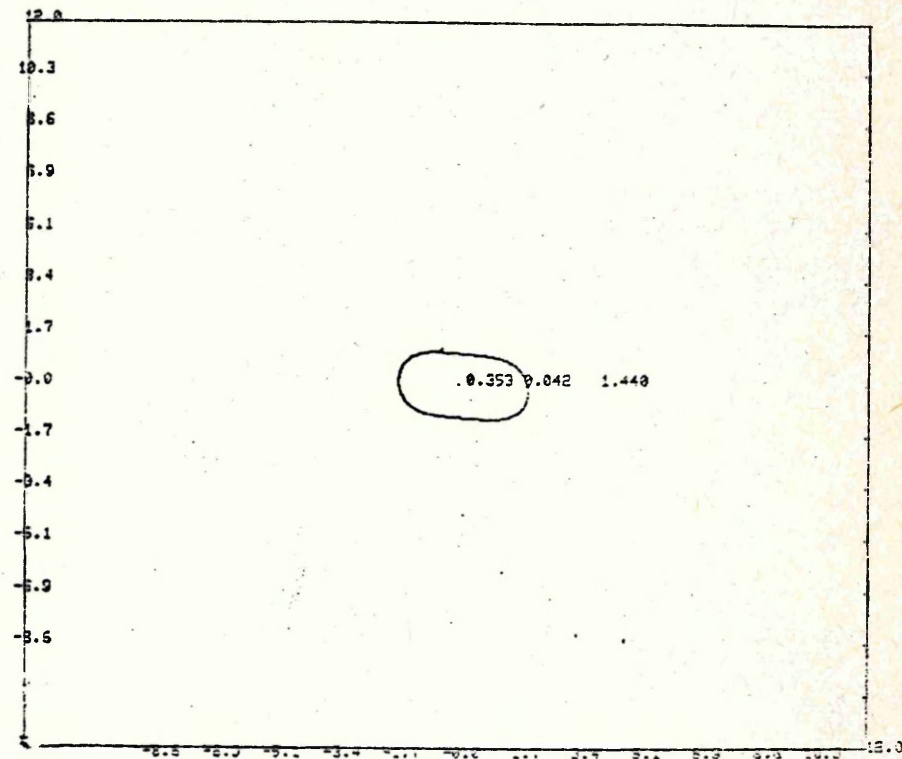
PRESS J FOR NEW PAGE & WINDOW
 PRESS L FOR NEW PAGE & XN
 PRESS ANY OTHER KEY FOR START
 WITH CURSOR, P...I, AS BEFORE

Fig. 5.27a

Contour on the
 Plane (x_1, x_2)
 with

$$x_3 = 0.0$$

$$x_4 = 0.0$$



PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C FOR F.P.METHOD
 PRESS D FOR 1,2,3RD CURVES

PRESS E FOR FUN. VAL. OUTPUT
 PRESS F FOR MIN. COOR. FINDING
 PRESS G,H,I FOR 2,3,4TH XN INPUT
 PRESS K FOR X-Y OUTPUT

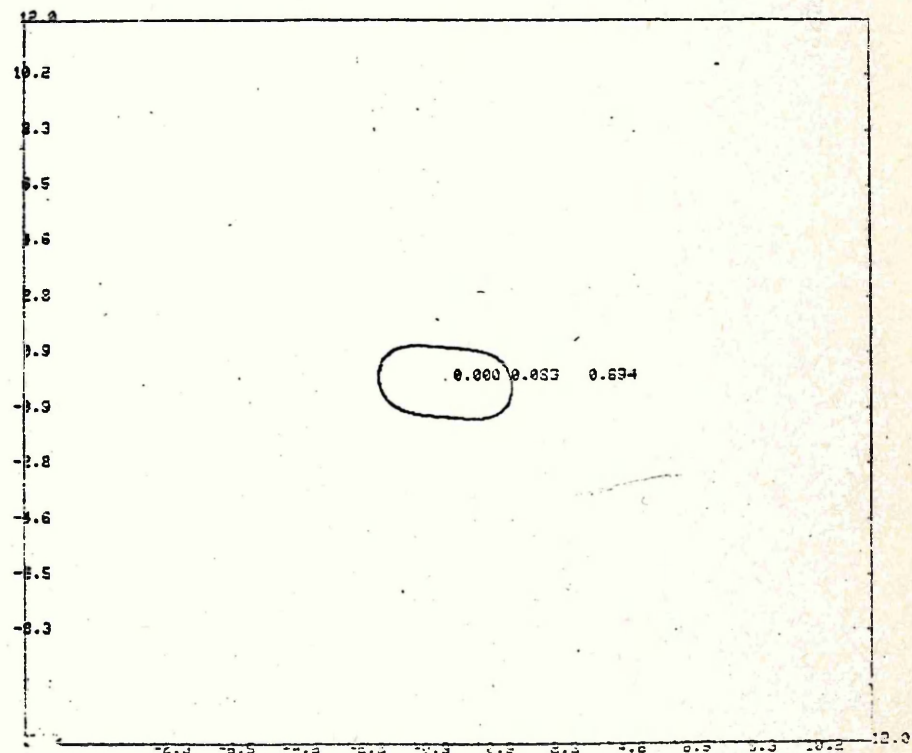
PRESS J FOR NEW PAGE & WINDOW
 PRESS L FOR NEW PAGE & XN
 PRESS ANY OTHER KEY FOR START
 WITH CURSOR, P...I, AS BEFORE

Fig. 5.27b

Contour on the
 Plane (x_1, x_2)
 with

$$x_3 = 1.0$$

$$x_4 = 1.0$$



CURVE FINISHED USE OPTIONS BELOW
 PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C TO DRAW NEW CURVE

PRESS D TO DRAW NEW CURVE
 PRESS E TO SEE OTHER OPTINS

GRAPH OF FUN. U DIST.

RESULT OF PROBLEM
 MIN. VAL. OF X_1, X_2, \dots
 0.124 0.063 0.175 0.175 0.9
 LAMBDA= 0.35 F= 0.663

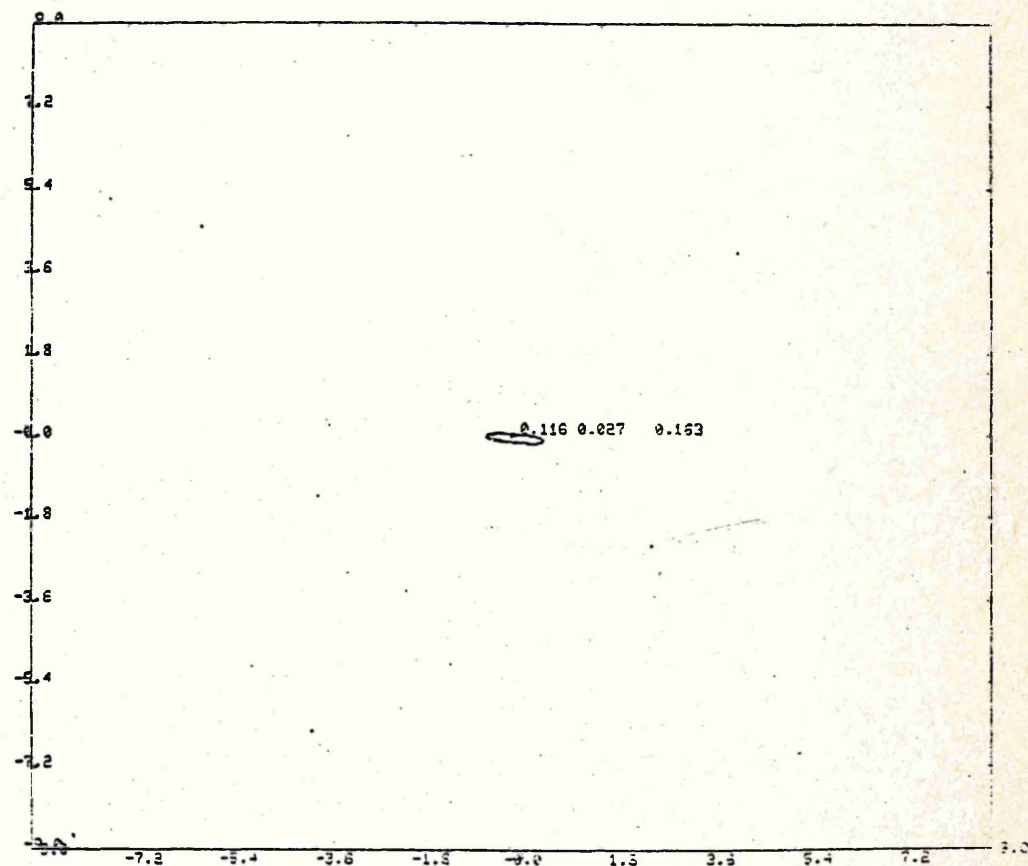
Fig. 5.27c
 Direct Search
 Curve.

PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C FOR F.P. METHOD
 PRESS D FOR 1,2,3RD CURVES

PRESS E FOR MIN. COOR FINDING
 PRESS G,H,I FOR 2,3,4TH XN INPUT
 PRESS K FOR X-Y OUTPUT

PRESS J FOR NEW PAGE & XN
 PRESS L FOR NEW PAGE & XN
 PRESS ANY OTHER KEY FOR START
 WITH CURSOR, A...I, AS BEFORE

Fig. 5.27d



CURVE FINISHED USE OPTIONS BELOW
 PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C TO DRAW NEW CURVUS

PRESS D TO DRAW NEW CURVE
 PRESS E TO SEE OTHER OPTINS

GRAPH OF FUN. U DIST.

RESULT OF PROBLEM
 MIN. VAL. OF X_1, X_2, \dots
 -0.093 0.014 -0.111 -0.111
 LANDM= 1.88 F= 0.005

Fig. 5.27e
 Min. value of
 function found
 by direct search.

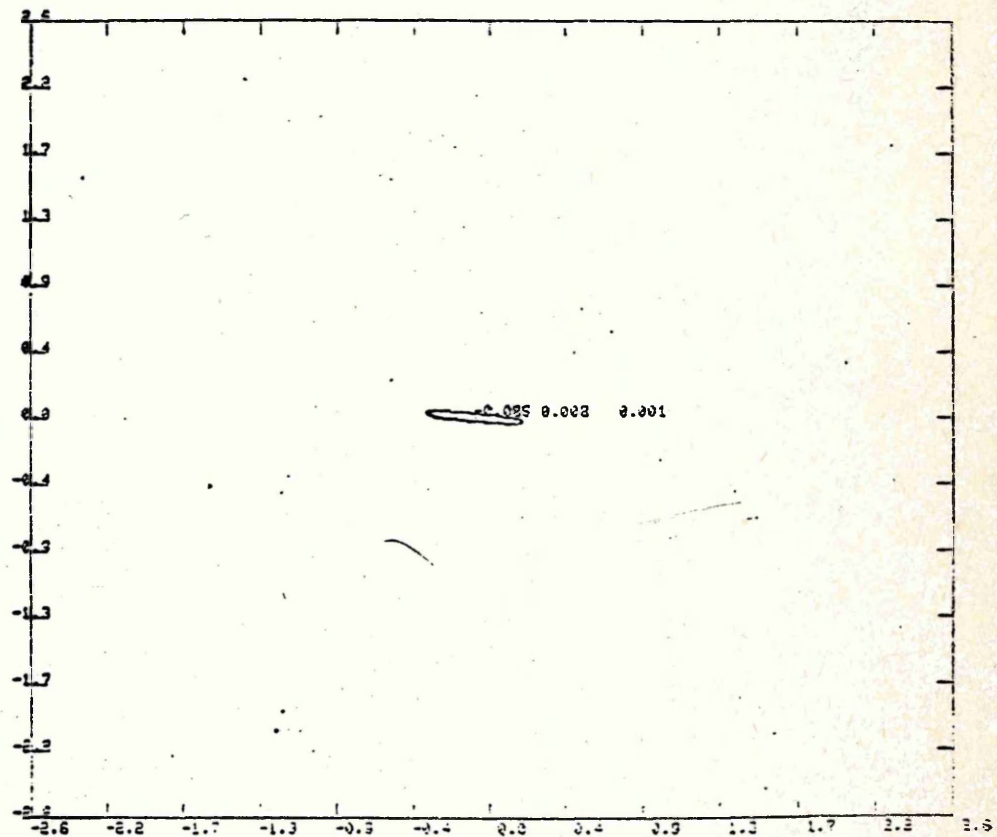
PRESS A TO FINISH
 PRESS B FOR CURSOR
 PRESS C FOR F.P.METHOD
 PRESS D FOR 1,2,3RD CURVES

PRESS E FOR FUN. VAL. OUTPUT
 PRESS F FOR MIN. VAL. FINDING
 PRESS G,H,I FOR 2,3,4TH XN INPUT
 PRESS K FOR X-Y OUTPUT

PRESS J FOR NEW PAGE & UNDO
 PRESS L FOR NEW PAGE & AN
 PRESS ANY OTHER KEY FOR START
 WITH CURSOR, A...I, AS BEFORE

Fig. 5.27f

A further
 reduced value
 of the function
 using the above
 value to plot
 the contour.



1.413 5.000 -5.000 1.413
 RANGE OF Y,FORMAT F7.3
 2.175 5.000 -5.000 2.175
 NON AXIAL DIMS,F7.3
 DIM 3
 0.000
 DIM 4
 0.000

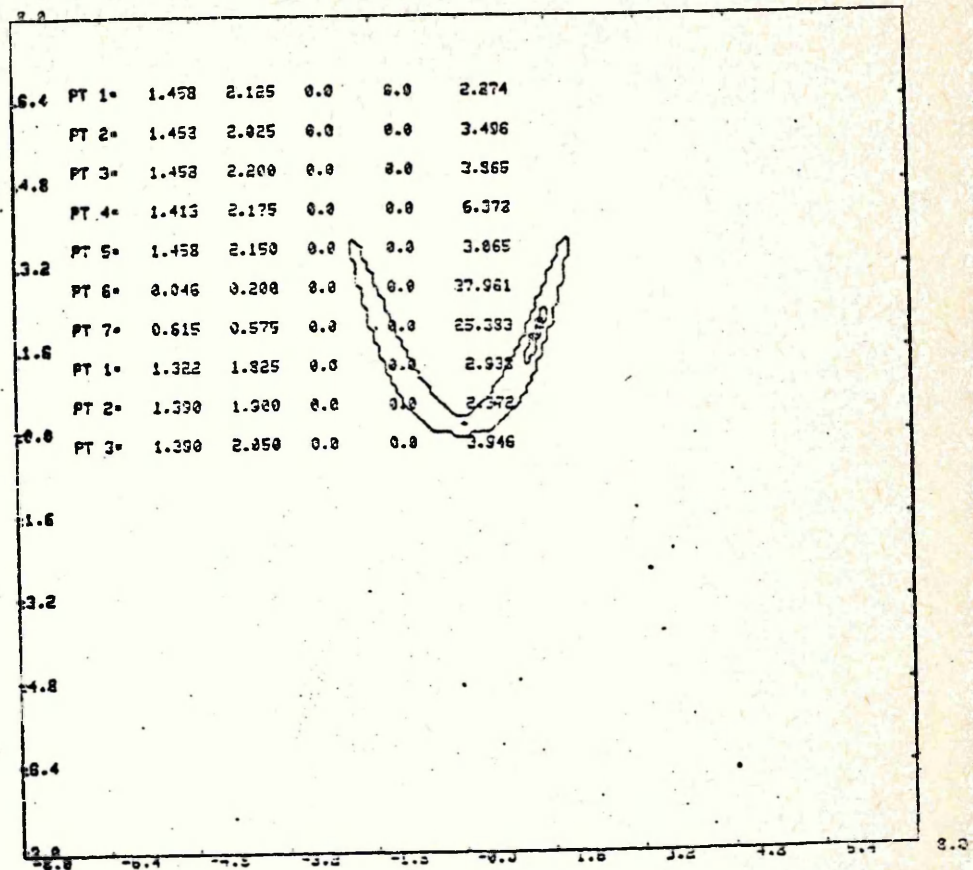
Fig. 5.28a

Contour on plane
 (x_1, x_2)

$$x_3 = 0.0$$

$$x_4 = 0.0$$

FMIN= 42.000
 FNC.EVAL. KN1= 560
 FMIN= 6.344
 FNC.EVAL. KN1= 730

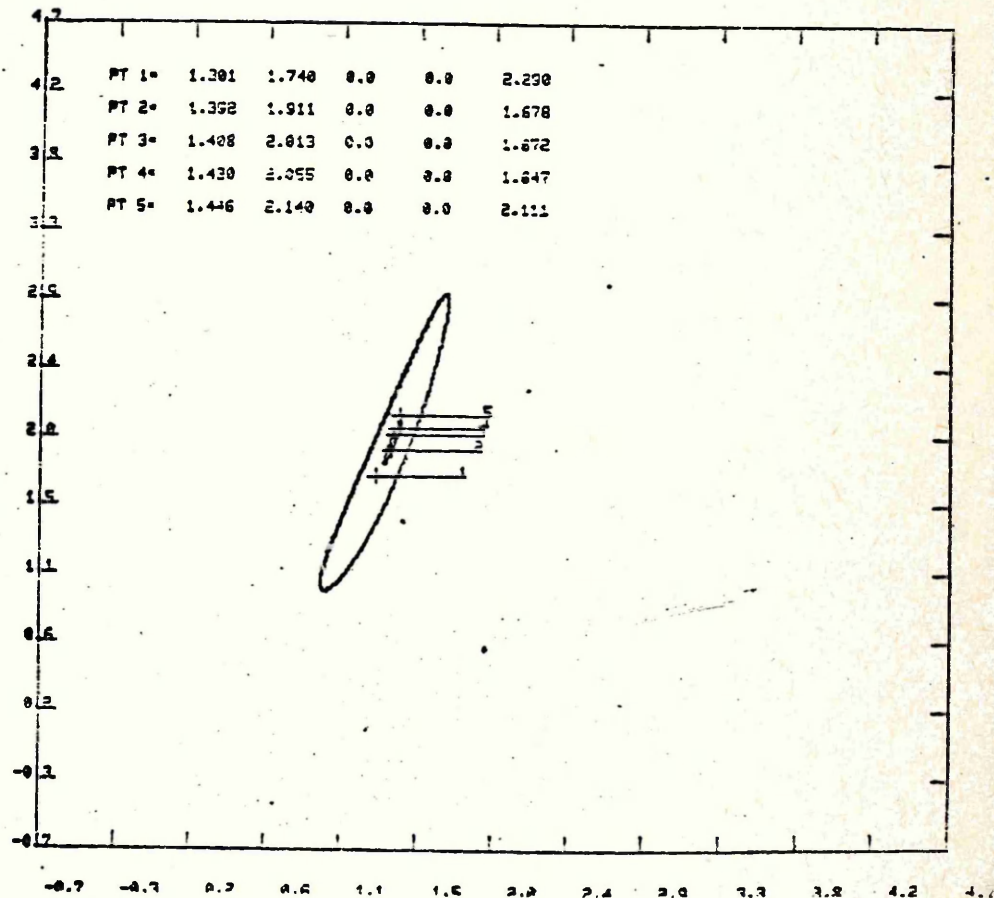


RANGE OF X,REAL
 1.408 5.0 -5.0 2.1408
 RANGE OF Y,REAL
 2.013 5.0 -5.0 2.013
 NON AXIAL DIMS,REAL
 DIM 3
 0.0
 DIM 4
 0.0
 VALUES CORRECT?
 Y

Fig. 5.28b

Zooming of Inner
 Contour.

FMIN= 11.100
 FNC.EVAL. KN1= 1216
 FMIN= 1.667
 FNC.EVAL. KN1= 292



PRESS C FOR CURVE FROM
PRESS D FOR 1,2,3RD CURVES

PRESS W,X,Y FOR 2,3,4TH XN INPUT
PRESS K FOR X-Y OUTPUT

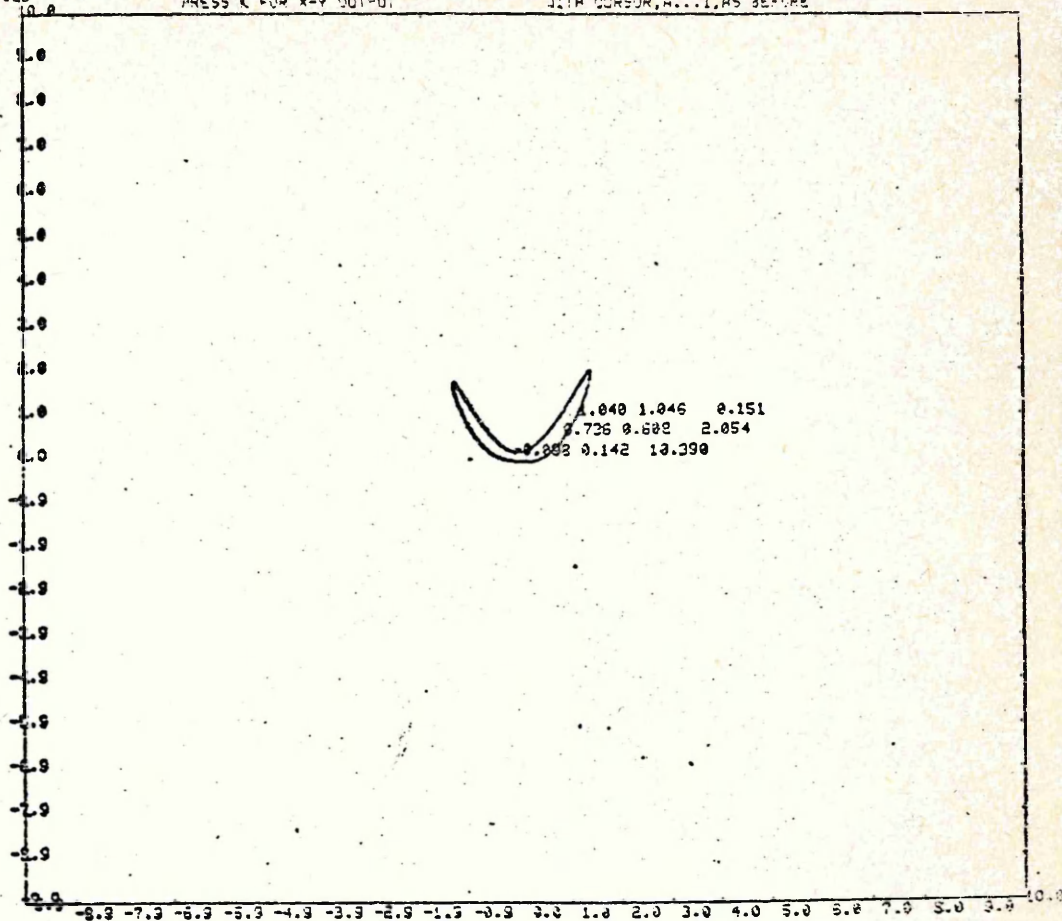
PRESS ANY OTHER KEY FOR START
WITH CURSOR, A...I, AS BEFORE

5.28c

Contour on
Plane (x_1, x_2)

$x_3 = 1.0$

$x_4 = 1.0$



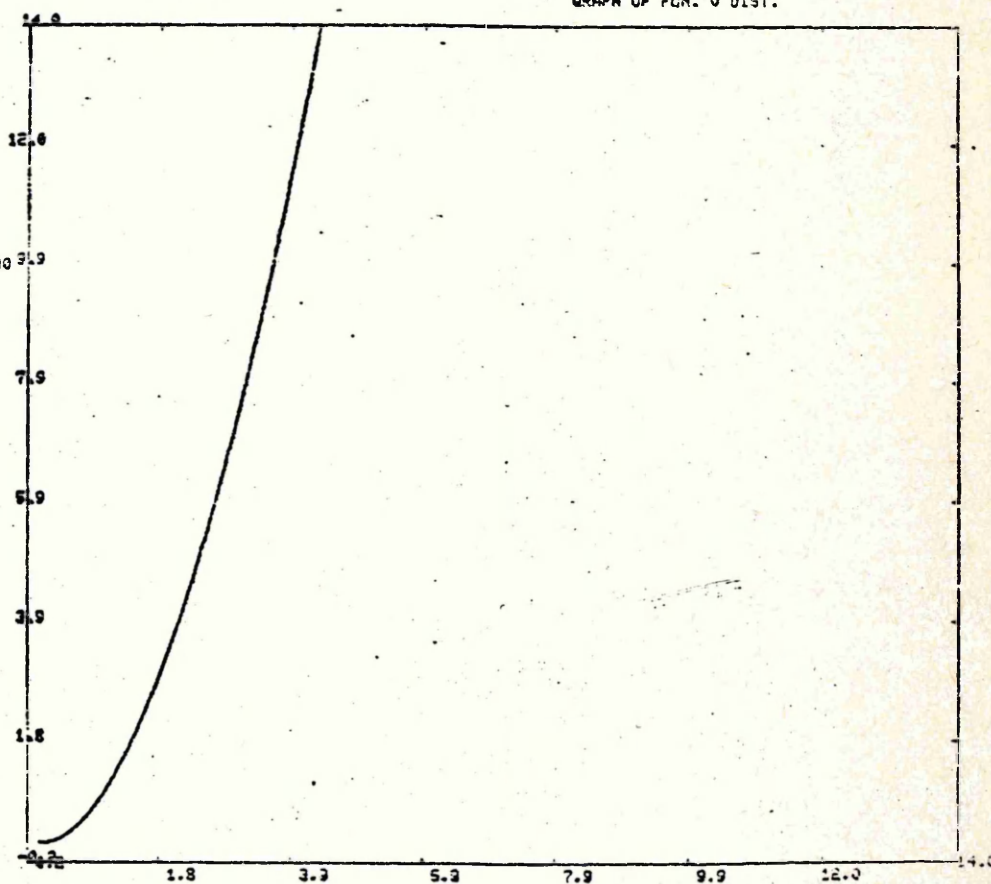
GRAPH OF FUN. U DIST.

RESULT OF PROBLEM

MIN. VAL. OF x_1, x_2
1.046 1.082 1.000 1.000
LAMDA= 0.06 F= 0.146

Fig. 5.28d

Direct Search
Curve.



PRESS A TO FINISH
PRESS E FOR CURSOR
PRESS C FOR F.F.METHOD
PRESS D FOR MIN COOR. FINDING
PRESS E,F FOR 1,2ND CONTOUR

PRESS G FOR NEW FUNCT.
PRESS H FOR NEW PAGE & METHOD
PRESS I STORE MIN.COOR. OF CONTOUR
PRESS J IF ONLY A NEW PAGE IS NEEDED
PRESS K TO CHANGE THE UINDC.

PRESS L TO CHANGE THE STEP SIZE
PRESS ANY OTHER KEY FOR START

WITH CURSOR,A...I,AS BEFORE

THIS PROBLEM DOES NOT HAVE
ANY CONSTRAINTS

2.0
1.8
1.5
1.3
1.0
0.8
0.5
0.3
0.0
-0.3
-0.5

PT 1= 0.993 1.005 1.000 1.000 0.003

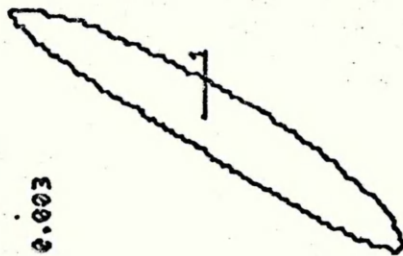
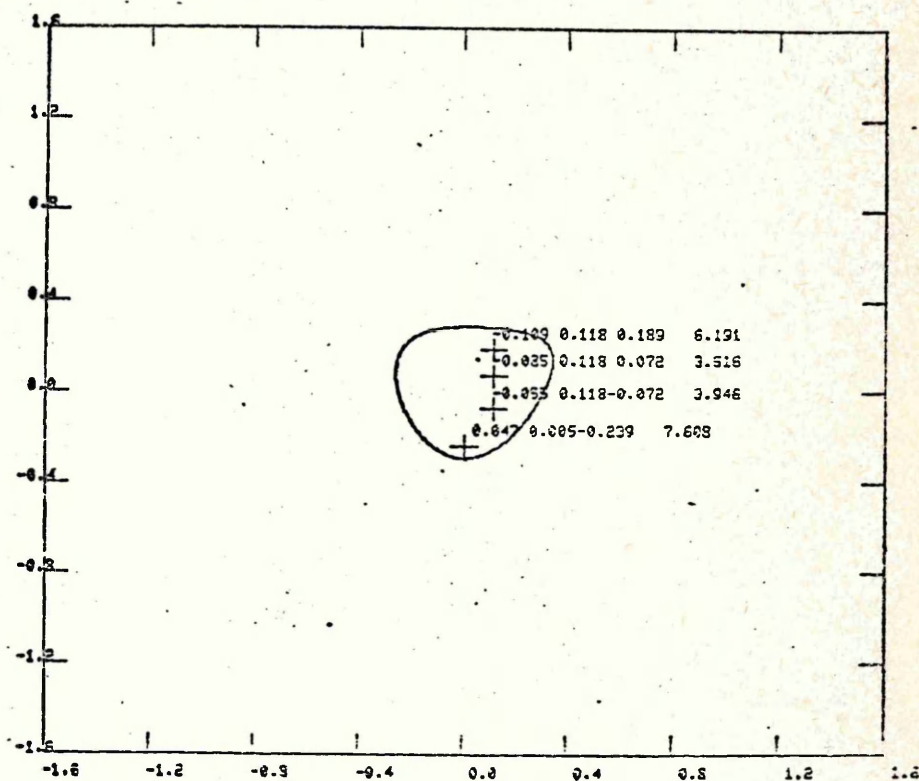


Fig. 5.29

Visual Location of the
Min. Value.

*MIN= 1.495
FNC.EVAL. KN1= 626

Fig. 5.30a
Contour on Plane
 (x_1, x_2) , $x_3 = 0.0$



Contour on plane
 (x_1, x_2) $x_3 = 1.0$

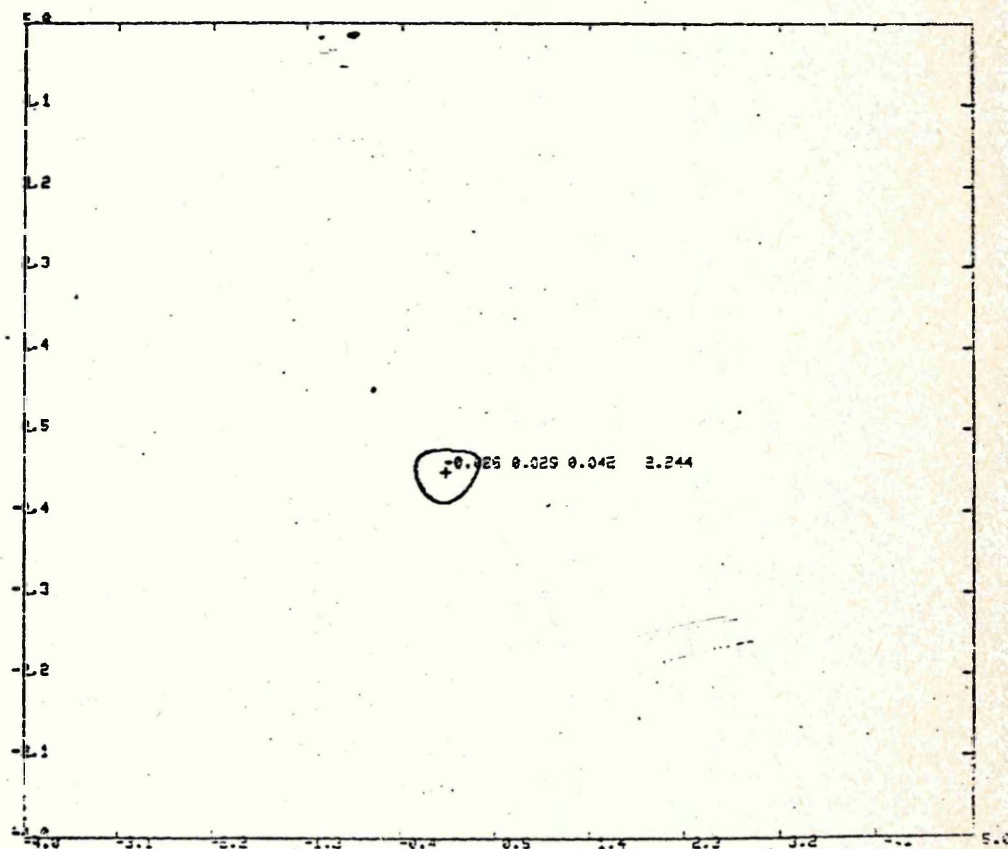
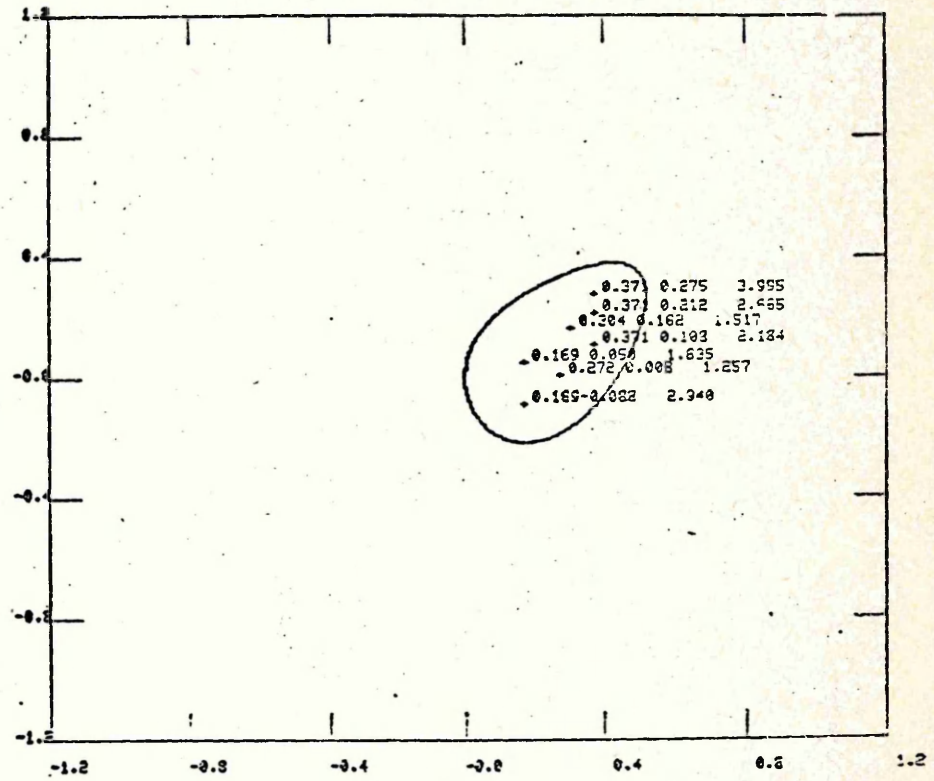


Fig. 5.30b



GRAPH OF FUN. U DIST.

RESULT OF PROBLEM
MIN. VAL. OF x_1, x_2, \dots
0.613 0.357 0.141
LAMBDA = 1.30 $F = 0.561$
DIRECTION VALUE = 0.02

Fig. 5.30c

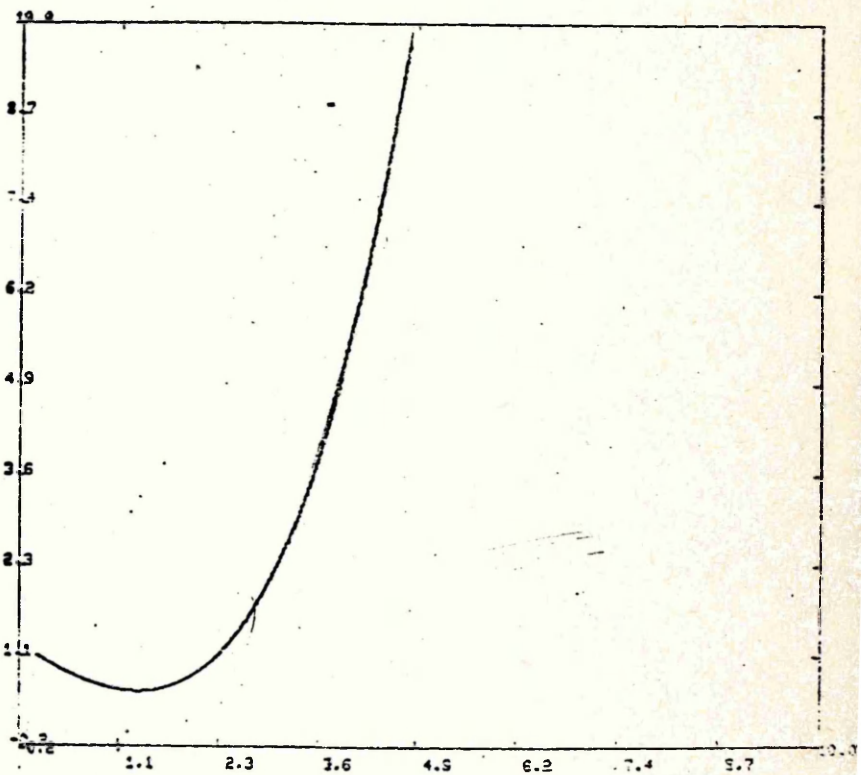
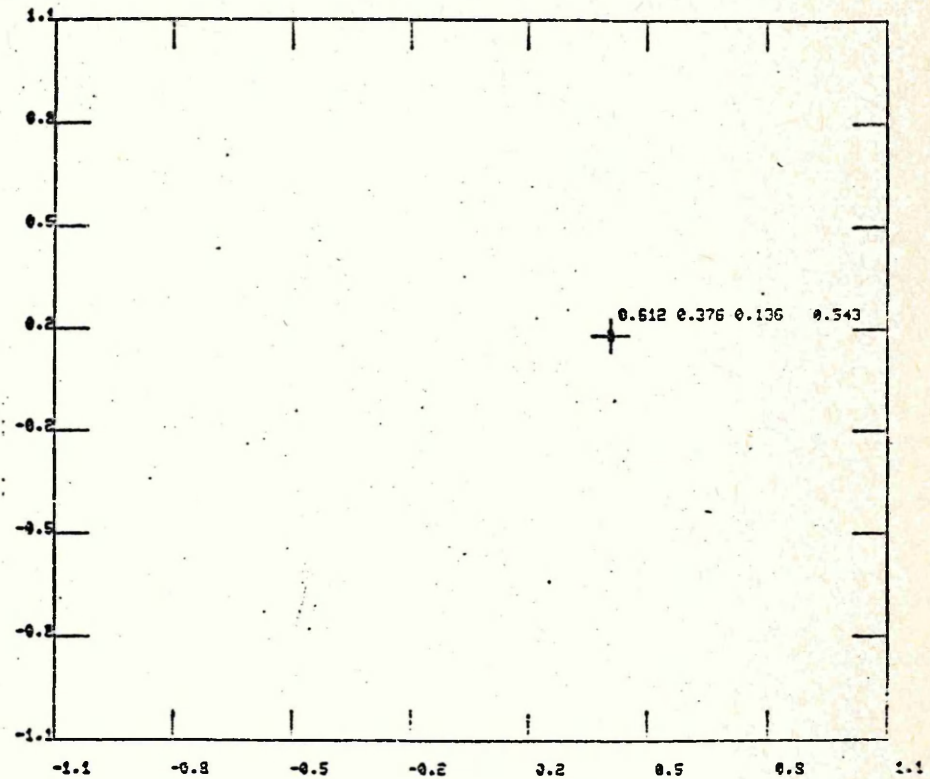


Fig. 5.30d



GRAPH OF FUN. U DIST.

RESULT OF PROBLEM
MIN. VAL. OF X_1, X_2, \dots
0.648 0.409 0.153
LAMBDA= 2.45 F= 0.509
DIRECTION VALLE= 0.31

Fig. 5.30e

Direct Search Curve.

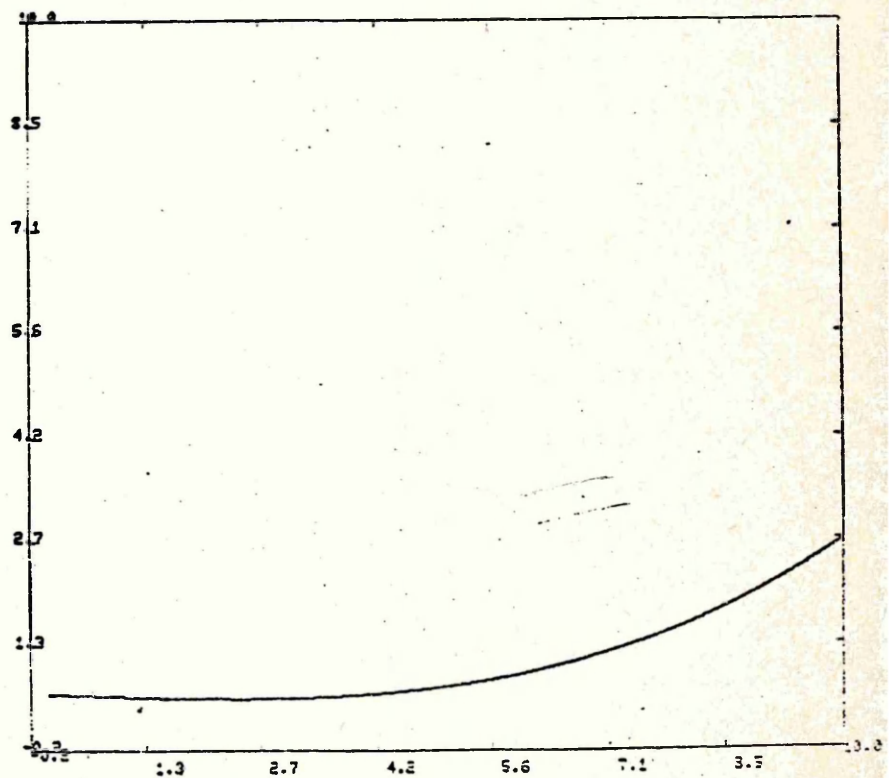
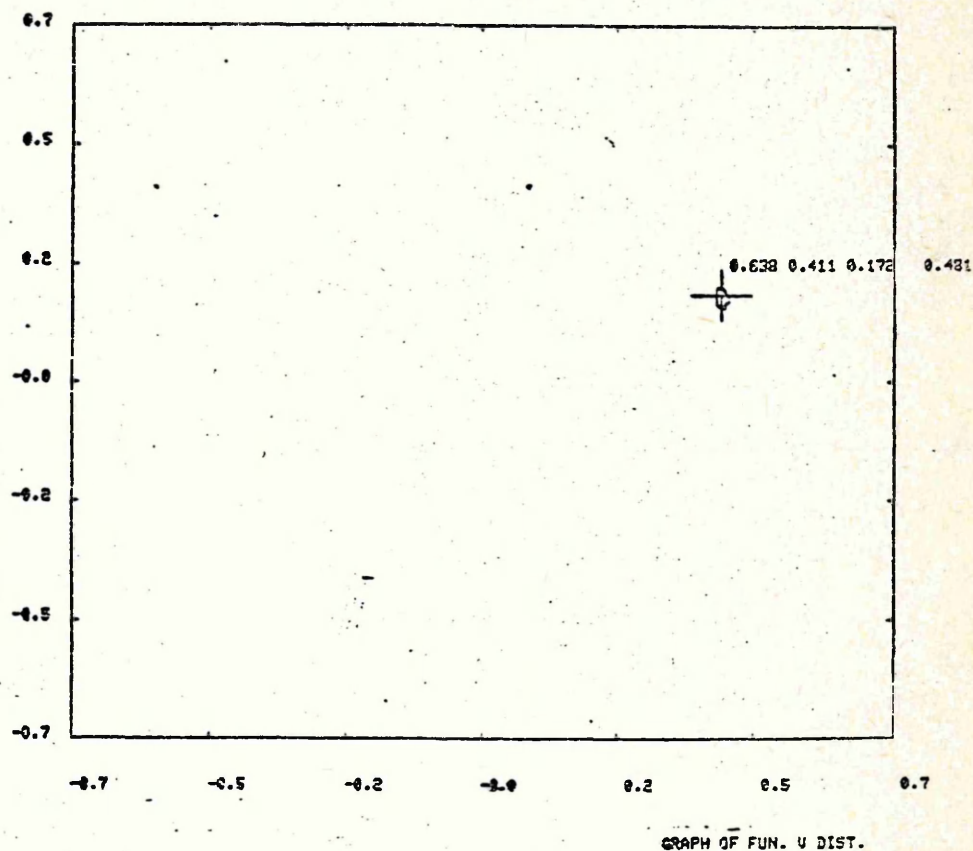


Fig. 5.30f



RESULT OF PROBLEM
MIN. VAL. OF X_1, X_2
0.317 0.632 0.420
LAMBDA= 7.50 F= 0.151
DIRECTION VALUE= 0.20

Fig. 5.30g

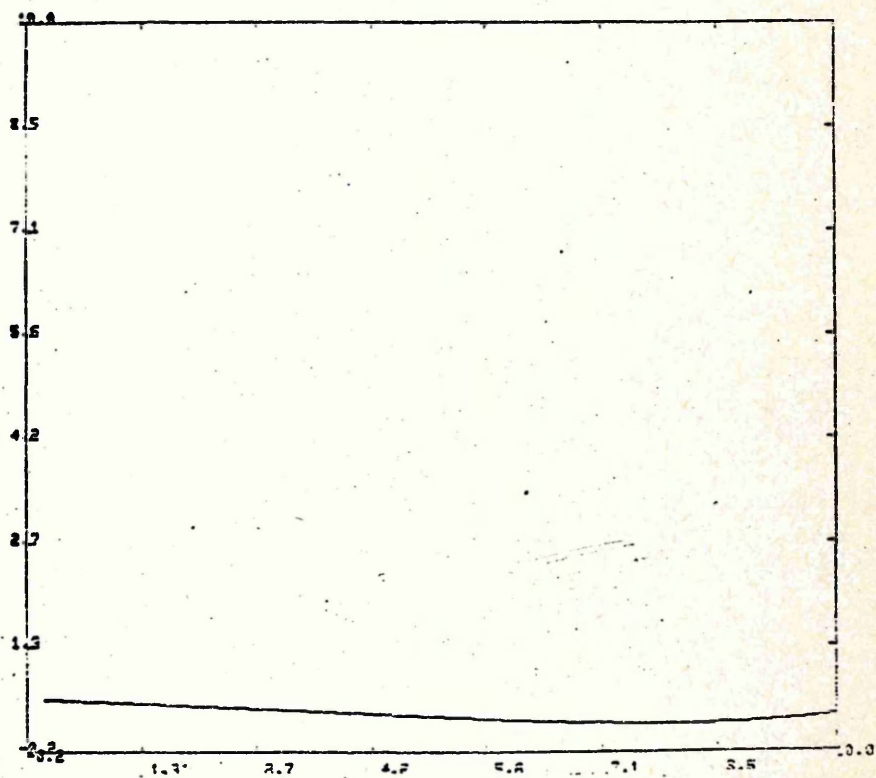
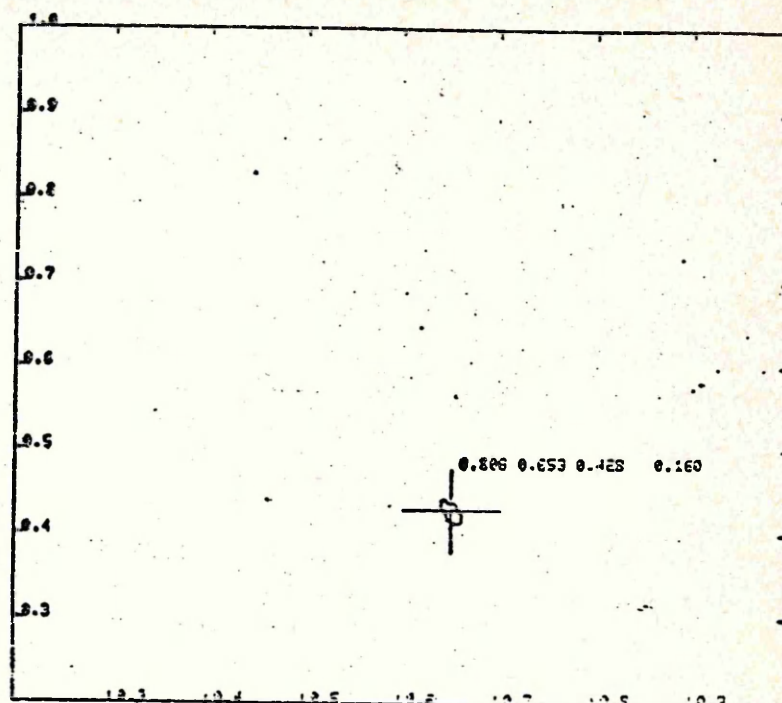
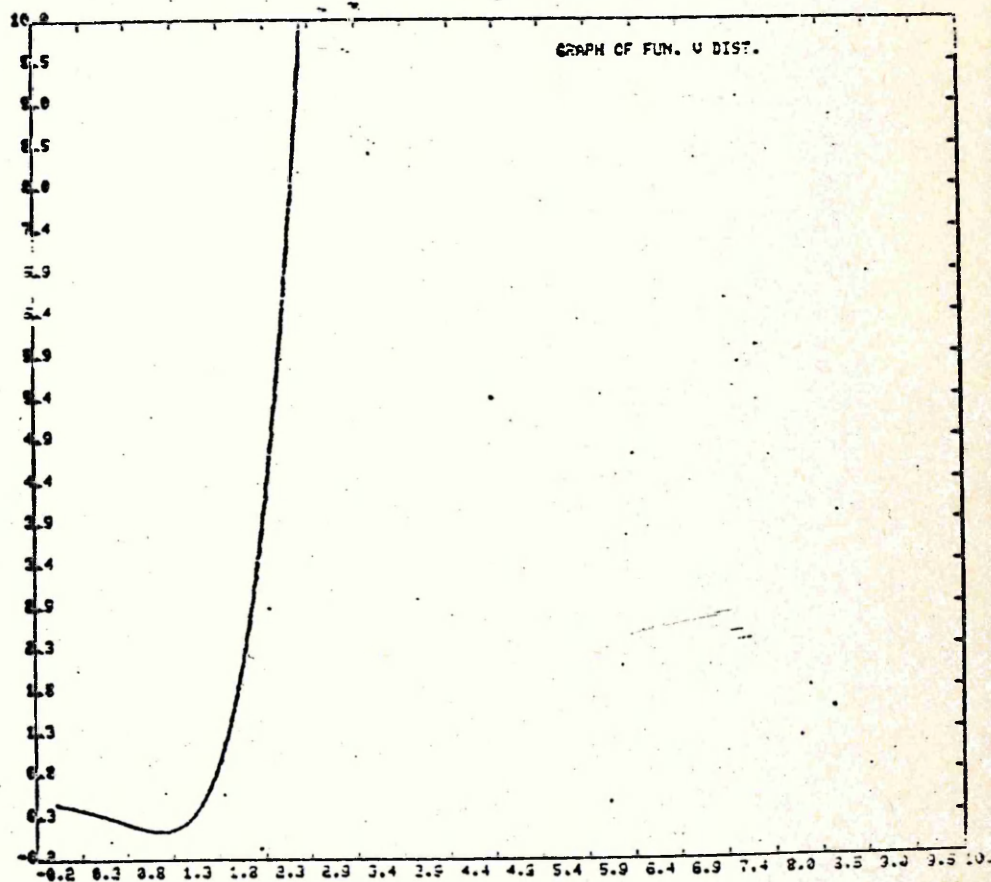


Fig. 5.30h



RESULT OF PROBLEM
MIN. VAL. OF χ^2
0.836 0.697 0.474
LAMBDA= 1.18 F= 0.132
DIRECTION VALUE= 0.21

Fig. 5.30i



RESULT OF PROBLEM
 MIN. VAL. OF X_1, X_2
 1.010 1.010 1.004 1.017 9.3
 LAMDA= 0.01 F= 0.024

Fig. 5.30j

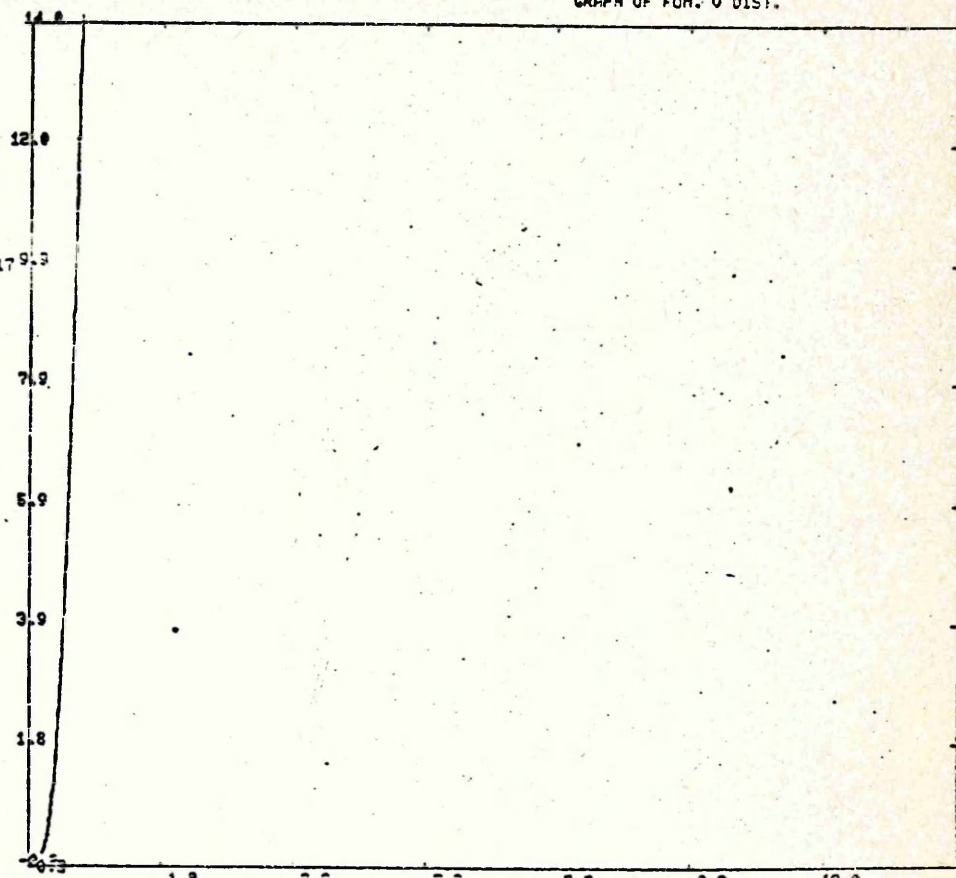


Fig. 5.30k

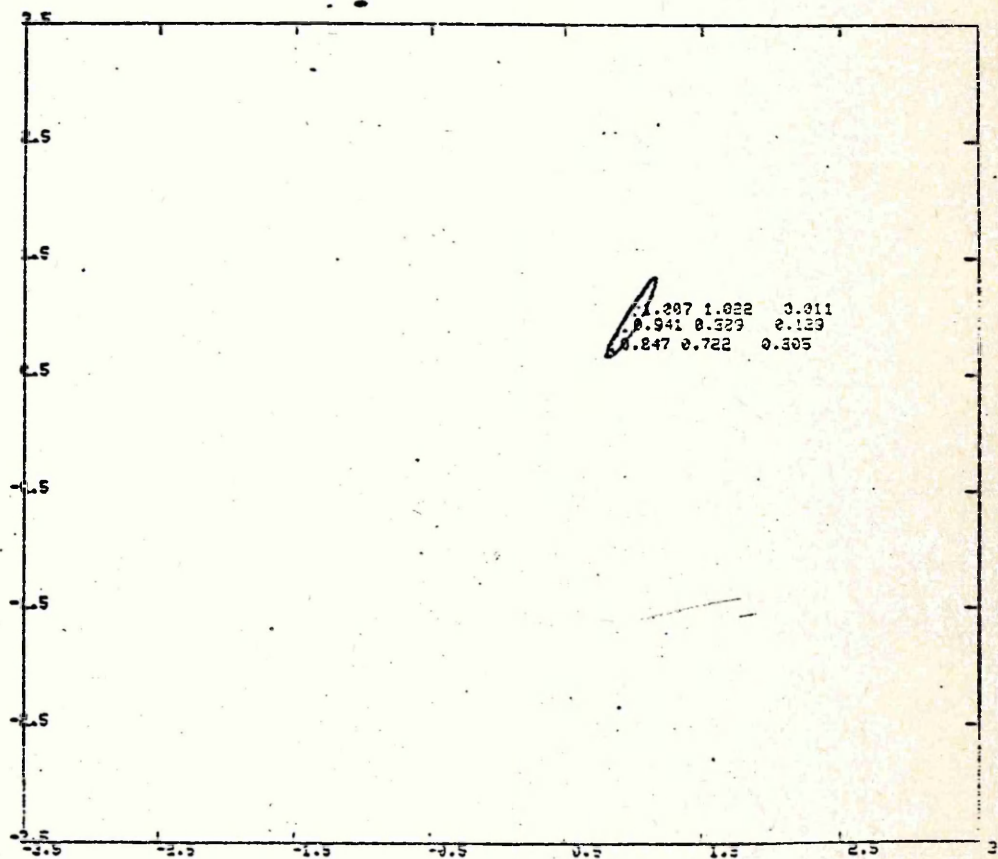
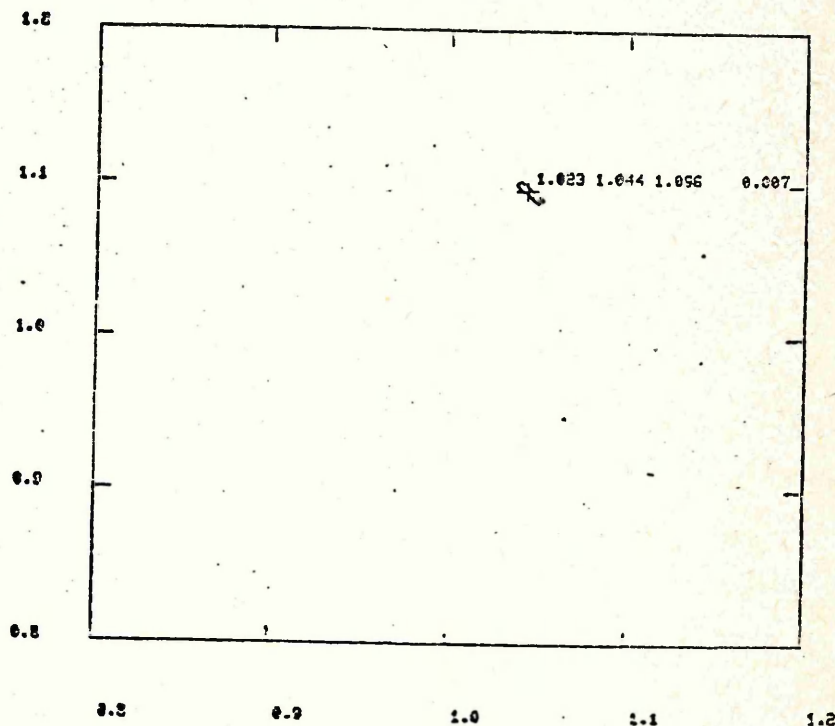


Fig. 5.301

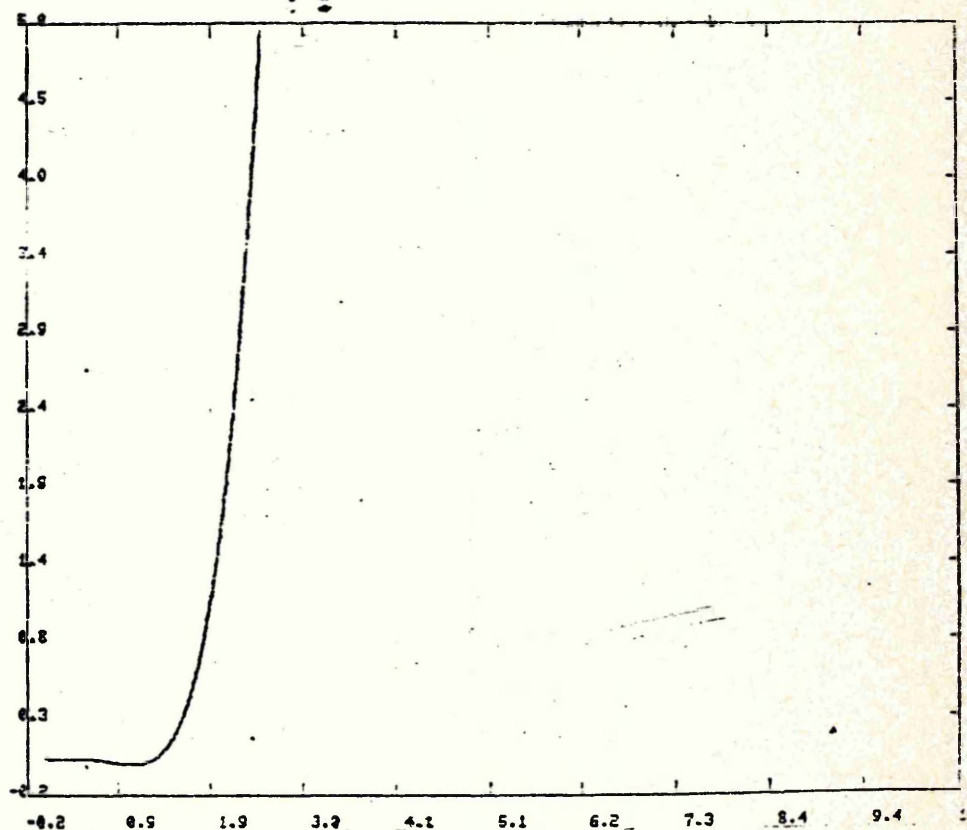
Min. located visually,
and further reduction
has been made by an
extra direct search.



RESULT OF PROBLEM
MIN. VAL. OF X_1, X_2, \dots
1.027 1.051 1.109
LAMDA= 1.03 F= 0.006
DIRECTION VALUE= 0.25

Fig. 5.30m

Final direct search.



A Practical Application

The previous chapter described how a series of test functions has been used to compare some of the optimization methods and to illustrate the facilities available on the graphics system.

In this chapter a description of how the system has been applied to a problem in molecular vibrational analysis is presented. This problem was chosen because it appeared to offer scope for optimization methods in general and because the graphical approach in particular also seemed to offer some promise. The problem had originally been attempted by a research chemist using an ad hoc approach.

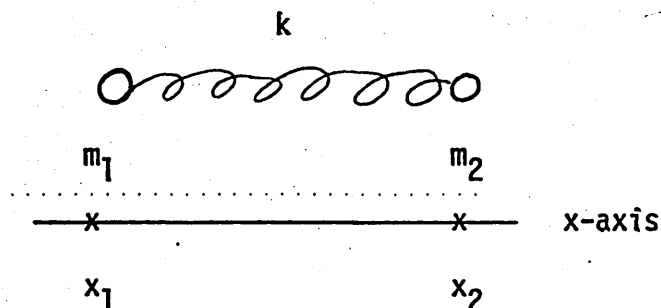
The background to the problem is first described and then the original approach and its drawbacks are discussed. This is followed by a description of how the existing programs were modified and how the problem was set up for solution by optimization methods and how it was then solved on the interactive system using a conventional direct search optimisation procedure.

A graphical approach to this problem is presented next, and the results are compared with those obtained by the other approaches.

Finally the merits of the graphical approach are discussed from the point of view of the user.

6.1 Molecular Vibrational Analysis

Consider a diatomic molecule constrained to move along the x-axis.



The molecule is free to move as a whole and each atom is free to move relative to the other atom, within the limits of the chemical bond which may be represented as a spring. The chemical bond exerts a force on each atom tending to restore the atoms to the position which is characteristic of the equilibrium bond length. Mathematically, the force is given by $F = -k(r - r_e)$, where 'r' represents the instantaneous bond length, given by $r = (x_2 - x_1)$, and r_e is the equilibrium bond length, $r_e = (x_2 - x_1)_e$. The atoms are sufficiently heavy that they obey 'classical mechanics' to a good degree of approximation; that is, the force on either atom causes a motion of the atom which responds to the force according to Newton's equation of motion $F_i = m_i a_i$, where F_i is the force acting on atom i of mass m_i , causing an acceleration a_i . This equation is only valid in a right hand cartesian co-ordinate system, which is not always convenient to use for molecules. It is a fairly simple matter to show for the above equation that an equivalent equation is the Lagrange Equation of motion:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{x}_i} \right) + \left(\frac{\partial V}{\partial x_i} \right) = 0 \quad i = 1, 2 \quad (1)$$

where T is the expression for the kinetic energy of the system, expressed as a function of velocities $\dot{x}_i = dx_i/dt = V_i$ and V is the potential energy expression for the system, expressed in terms of the atom positions x_i . The expressions for T and V are:

$$2T = m_1 \dot{x}_1^2 + m_2 \dot{x}_2^2, \text{ and} \quad (2)$$

$$V = \frac{1}{2}k \{ (x_2 - x_{2e}) - (x_1 - x_{1e}) \}^2 \quad (3)$$

It is now convenient to define cartesian displacement co-ordinates, which represent the displacement of the atoms from their positions corresponding to the equilibrium configuration of the molecule, by $x_i = (x_i - x_{ie})$. Since x_{ie} represents a constant $\dot{x}_i^0 = \dot{x}_i^0$, and the kinetic and potential energies can be written in terms of the displacements co-ordinates as:

$$2T = m_1 \dot{x}_1^2 + m_2 \dot{x}_2^2 \quad \text{and} \quad 2V = k(x_2 - x_1)^2 \quad (4)$$

Now applying the Lagrange Equation of Motion, equation(1) to both x_1 and x_2 :

$$\frac{\partial T}{\partial x_1} = m_1 \dot{x}_1, \quad \frac{d}{dt} \frac{\partial T}{\partial \dot{x}_1} = m_1 \ddot{x}_1 \quad \text{where } \ddot{x}_1 \text{ represents the second}$$

derivatives of x_2 with respect to time.

$$\frac{\partial V}{\partial x_1} = k(x_1 - x_2) \quad \text{therefore,}$$

$$0 = m_1 \ddot{x}_1 + kx_1 - kx_2 \quad \text{and} \quad 0 = m_2 \ddot{x}_2 + kx_2 - kx_1 \quad (5)$$

The differential equations (5) are infinite in number, but it turns out that there is a solution for x_1 and x_2 which is simple and in terms of which any other solution can be written as a linear combination. This particular set of solutions for x_1 and x_2 is called the 'Normal Mode Solution', the motion is simple in that atoms 1 and 2 move in-phase, and with the same frequency of harmonic oscillation about the equilibrium positions. Mathematically, the solutions are of the form:

$$x_i = A_i \sin(\lambda^{\frac{1}{2}} t + \psi) \quad (6)$$

where A_i is the maximum amplitude of displacement from equilibrium position for atom i , $\lambda = 4\pi^2 V^2$ and V is the frequency of vibration and ψ is the phase of the vibration. In order to substitute equation (6) into equation (5) the derivatives must be calculated. The derivatives are:

$$x_i = \lambda^{\frac{1}{2}} A_i \cos(\lambda^{\frac{1}{2}} t + \psi) \quad ; \quad \ddot{x}_i = -\lambda A_i \sin(\lambda^{\frac{1}{2}} t + \psi)$$

Now substituting into equation (5)

$$\begin{aligned} 0 &= -m_1 \lambda A_1 \sin(\lambda^{\frac{1}{2}} t + \psi) + k(A_1 - A_2) \sin(\lambda^{\frac{1}{2}} t + \psi) \\ 0 &= -m_2 \lambda A_2 \sin(\lambda^{\frac{1}{2}} t + \psi) + k(A_2 - A_1) \sin(\lambda^{\frac{1}{2}} t + \psi) \end{aligned} \quad (7)$$

Since these equations must be valid at all times t , the $\sin(\lambda^{\frac{1}{2}} t + \psi)$ terms all can be cancelled, leaving two simultaneous equations, known as normal equations;

$$\begin{aligned} 0 &= (k - m_1 \lambda) A_1 - k A_2 \\ 0 &= (k - m_2 \lambda) A_2 - k A_1 \end{aligned} \quad (8)$$

Equations (8) have non-trivial solutions if and only if the determinant of the A_i 's vanishes, that is if

$$m_1 m_2 \lambda^2 - (m_1 + m_2) k \lambda = 0 \quad (9)$$

whose roots are; $\lambda_1 = 0$ and $\lambda_2 = (m_1 + m_2) k / m_1 m_2$.

The kinetic and potential energy expressions can be written in matrix form:

$$2T = \begin{Bmatrix} \dot{x}_1 & \dot{x}_2 \end{Bmatrix} \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \dot{\underline{x}}^t \underline{I} \dot{\underline{x}} \quad (10)$$

$$2V = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underline{x}^t \underline{F} \underline{x} \quad (11)$$

where \underline{I} is the kinetic energy matrix and \underline{F} is the potential energy matrix. The vibrational secular equation for the system can now be defined as:

$$|\underline{F} - \lambda \underline{I}| = 0 \quad (12)$$

where $\lambda = 4\pi^2\nu^2$ and a scalar.

Equation (10) when solved yields the eigenvalue λ_i . Since each eigenvalue λ_i has a set of amplitudes associated with it, the amplitudes can be referred to as A_{i1} and A_{i2} . For the diatomic system, $\lambda_1 = 0$ and $A_{11}/A_{12} = 1$ and $\lambda_2 = k/w$ and $A_{21}/A_{22} = -m_2/m_1$.

The two normal equations must be simultaneously satisfied for each eigenvalue λ_1 and λ_2 , and therefore is it necessary to have the set of equations below which arise from associating the appropriate amplitudes with each eigenvalue:

$$\begin{aligned} kA_{11} - kA_{12} &= m_1\lambda_1A_{11} & kA_{21} - kA_{22} &= m_1\lambda_2A_{21} \\ &\text{and} & & \\ -kA_{11} + kA_{12} &= m_2\lambda_1A_{12} & -kA_{21} + kA_{22} &= m_2\lambda_2A_{22} \end{aligned} \quad (13)$$

A diagonal eigenvalue matrix is now defined as:

$$\underline{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad (14)$$

and an eigenvector matrix, with i^{th} column elements proportional to the A_i amplitudes for the corresponding eigenvalue λ_i ,

$$\underline{L} = \begin{bmatrix} e_1 A_{11} & e_2 A_{12} \\ e_1 A_{12} & e_2 A_{22} \end{bmatrix} \quad (15)$$

The above four equations can be written as:

$$\underline{F} \underline{L} = \underline{I} \underline{L} \underline{\Lambda} \quad (16)$$

If equation (16) is multiplied by (\underline{L}^{-1}) on both sides, then,

$$\underline{L}^{-1} \underline{F} \underline{L} = \underline{L}^{-1} \underline{I} \underline{L} \underline{\Lambda} \quad (17)$$

where a matrix product of $\underline{L}^{-1} \underline{F} \underline{L}$ is called a unitary transformation of the matrix \underline{F} .

If $\underline{L}^{-1} \underline{I} \underline{L} = \underline{E}$, then $\underline{L}^{-1} \underline{F} \underline{L} = \underline{\Lambda}$, and the problem of finding the frequencies (i.e., $\underline{\Lambda}$) and normal modes (i.e. amplitudes or \underline{L}) reduces to finding a unitary transformation that will simultaneously diagonalize \underline{I} and \underline{F} , \underline{I} is a diagonal \underline{E} or unit matrix, and \underline{F} matrix to the diagonal $\underline{\Lambda}$ matrix.

A more usual method of treating equation (16) is to multiply the left hand side by $\underline{I}^{-1} = \underline{G}$ to obtain

$$\underline{I}^{-1} \underline{F} \underline{L} = \underline{I}^{-1} \underline{I} \underline{L} \underline{\Lambda} \text{ or } \underline{G} \underline{F} \underline{L} = \underline{L} \underline{\Lambda} \quad (18)$$

This equation is known as the Wilson GF equation, where \underline{G} is the inverse kinetic energy matrix and the equation above is equivalent to the determinantal equation:

$$|\underline{G} \underline{F} - \lambda \underline{E}| = 0 \quad (19)$$

Equation (18) is multiplied by \underline{L}^{-1} on the left hand side to obtain

$$\underline{L}^{-1} (\underline{G} \underline{F}) \underline{L} = \underline{\Lambda} \quad (20)$$

which indicates that the solution is accomplished by finding a matrix \underline{L} which will diagonalize the product matrix $\underline{G} \underline{F}$ by a unitary transformation. By such a diagonalization both eigenvalues ($\underline{\Lambda}$) and eigenvectors (\underline{L}) are obtained, which correspond to the frequencies and normal modes of the vibrations of the molecular system, respectively. One problem arises in the above process whereas both \underline{F} and \underline{G} are symmetric matrices (i.e., $f_{ij} = f_{ji}$ and $g_{ij} = g_{ji}$), the product $\underline{G} \underline{F}$ is not symmetric, and special techniques are needed to accomplish the diagonalization.

Recently, a more efficient method of performing vibrational analysis, which involves a single matrix diagonalization has been introduced by W. D. Gwinn (45), and since most of the computer time is involved with the matrix diagonalization step, this method takes roughly half the computer time as the older Wilson $\underline{G} \underline{F}$ matrix method which needs two diagonalizations. In order to understand this method, some consideration should be given to the co-ordinate systems in which the kinetic and potential energy expressions are expressed.

The kinetic energy is expressed in Cartesian co-ordinates as already described. Briefly the expression consists of the sum of $\frac{1}{2} mV^2$ terms, three for each atom since each can move in general in three dimensions, thus \underline{I} is generally of order $3N \times 3N$ for an N -atom molecule.

The potential energy is generally not expressed in Cartesian co-ordinates. Instead, the co-ordinates which have been found to be most useful for changes in potential energy as a molecule vibrates are called 'internal displacement co-ordinates'. These consist of stretching of chemical bonds, bending of valence angles, out of plane bending, and torsion about non-terminal bonds. Other types of co-

ordinates are sometimes defined, but the above are sufficient to completely describe molecular vibrations. Usually there are $3N - 6$, or $3N - 5$ for a linear molecule, internal co-ordinates, since movement and rotation are not internal motions which change internal distances in molecules, but are rather external motions of the molecules as a whole.

The potential energy is generally taken to be a sum of quadratic terms in the internal co-ordinates, R_j , $j = 1, 2, \dots, 3N - 6$, therefore

$$2V = \sum_{i,j} f_{ij} R_i R_j = \underline{R}^T \underline{F} \underline{R}$$

$$\text{where } \underline{R} = R_1, R_2, \dots, R_{3N-6} \quad (21)$$

The kinetic energy expression in Cartesian co-ordinates is:

$$2T = \underline{\dot{X}}^T \underline{M} \underline{\dot{X}} \quad (22)$$

Since the potential energy matrix \underline{F} is of order $(3N - 6 \times 3N - 6)$, while the kinetic energy matrix \underline{M} is $(3N \times 3N)$, they cannot be added or subtracted to give the vibrational secular equation. There are two ways of proceeding, either change the kinetic energy matrix \underline{I} to represent internal motion in terms of co-ordinates \underline{R} , or to change the potential energy matrix \underline{F} to represent forces required to stretch bond, trend angles etc. The first alternative leads to the Wilson G F matrix method.

The second approach is the Gwinn method which can be described as follows. The mass-weighted co-ordinates are defined by $q_i = (m_i)^{\frac{1}{2}} X_i$ or $\underline{q} = \underline{M}^{\frac{1}{2}} \underline{X}$. If left hand side is multiplied by $\underline{M}^{-\frac{1}{2}}$ it gives

$\underline{X} = \underline{M}^{-\frac{1}{2}} \underline{q}$ or $\underline{\dot{X}} = \underline{M}^{-\frac{1}{2}} \underline{\dot{q}}$ and $\underline{\dot{X}}^t = \underline{\dot{q}}^t \underline{M}^{-\frac{1}{2}}$. The last expression follows from the fact that \underline{M} is a diagonal matrix of masses, and hence $\underline{M}^{-\frac{1}{2}}$ is a diagonal matrix containing $1/m_i^{\frac{1}{2}}$ as diagonal elements, and the transpose of a diagonal matrix is the same matrix. Thus in terms of mass weighted Cartesian co-ordinates the kinetic energy becomes:

$$2T = \underline{\dot{X}}^t \underline{M} \underline{\dot{X}} = \underline{\dot{q}}^t \underline{M}^{-\frac{1}{2}} \underline{M} \underline{M}^{-\frac{1}{2}} \underline{\dot{q}} = \underline{\dot{q}}^t \underline{E} \underline{\dot{q}} = \underline{\dot{q}}^t \underline{\dot{q}}$$

i.e., $\underline{T} = \underline{E}$ in mass-weighted cartesian co-ordinates, since the result of pre and post multiplying \underline{M} by $\underline{M}^{-\frac{1}{2}}$ is to form a unit matrix.

The potential energy matrix is also converted to mass-weighted co-ordinates as follows:

$$2V = \underline{R}^t \underline{F} \underline{R} = \underline{X}^t \underline{B}^t \underline{F} \underline{B} \underline{X} = \underline{q}^t (\underline{M}^{-\frac{1}{2}} \underline{B}^t \underline{F} \underline{B} \underline{M}^{-\frac{1}{2}}) \underline{q} = \underline{q}^t \underline{V} \underline{q}$$

Thus, in Gwinn's approach, the force constants for internal co-ordinates are read in and converted by a matrix product above, involving matrix \underline{B} and the atomic masses, to force constants appropriate for mass-weighted Cartesian co-ordinates, indicated by elements V_{ij} . These may be thought of as force constants for moving each atom in the x, y, z direction, divided by the mass of the atom.

Since $\underline{T} = \underline{E}$ and $\underline{F} = \underline{V}$ in mass-weighted Cartesian co-ordinates, the equation (12) becomes:

$$|\underline{V} - \lambda \underline{E}| = 0$$

or equation (16) becomes,

$$\underline{V} \underline{L} = \underline{L} \underline{\Lambda}$$

i.e. matrix \underline{V} can now be diagonalized by $\underline{L}^{-1} \underline{V} \underline{L} = \underline{\Lambda}$. The value of \underline{V} is often used to produce calculated frequencies.

6.2 The Minimization Problem

Given certain input data it is required to manipulate the values of a set of force constants in order to achieve a set of calculated frequencies which are as close as possible to a corresponding set of observed frequencies.

When the calculated frequencies are found the next step is to compute, at each temperature, the vibrational entropy and the contribution to the total vibrational entropy from each calculated and observed frequency. These two values are used to obtain the sum of squares of weighted error. This is taken as the criterion of fit between the two sets of frequencies and is to be minimized.

To start the procedure data on molecular geometry and specifications must be provided. The existing programs require as input:-

- (1) Atom co-ordinates, which may be either cartesian co-ordinates, or more usually the polar co-ordinates of each atom, which are written in terms of the bond distances and angles of symmetry.

The input for each atom consists of 'I', the atom number; 'JJ' the position of the origin, 'KK', the polar axis; 'R', 'e', 'φ', the polar co-ordinates of the atom; 'LL', a rotation code; 'NAX' the axis about which the rotation is to be carried out, and 'ALPHA' is the angle of rotation.

- (2) NAT; number of atoms.
- (3) NLBE; number of linear angle bonding.
- (4) WT; atomic weight of atoms.
- (5) NANG; number of angles to be calculated.
- (6) NFK; number of force constants.
- (7) INDEX; this integer assigns a particular value to a certain pair of vibrational co-ordinates.

(8) F_K ; the force constants.

All of the input values except the force constants are fixed for any given molecule. The values of the force constants have a direct effect on the calculation of the computed frequencies, which results in the reduction or the increase of the sum of squares of the weighted error. Therefore, the initial values assigned to force constants are of great importance, and they can only vary between zero and given upper limits. These points are analyzed in more detail in the next section.

6.3 Methods of Solution

Three approaches have been compared, using the molecule ethane, which produces 18 frequencies from 5 force constants.

- (i) 'Hit and miss' approach.
- (ii) Conventional direct search optimization method.
- (iii) Graphical plane analysis method.

6.3.1 Hit and miss approach

This problem was originally being dealt with by a research chemist using a 'hit and miss' approach. That is by altering the initial values of force constants, one at a time, and observing any changes in the sum of squares of weighted errors. If the error was reduced, then the force constant was kept at this value, the next one altered and the result examined. This process could, in principle, be carried out until a desirable result was obtained. The disadvantage of this approach was that it was very time consuming, especially when conducted in batch computing mode. Also, most force constants, when altered, tend to change frequencies of the same symmetry. Therefore,

although one frequency may be changed to a value closer to its respective observed frequency, the other frequencies will be affected by this change and their value will be increased, resulting in no change or an increase of the weighted error.

With the hit and miss approach the change of force constants resulted in a value of the error where any further reduction in the size of the error was found to be difficult to obtain. The value for the error and respective force constants for ethane were:-

Sum of squares of weighted error (SSQWR) = 37093.5430, with the force constants

$$\begin{aligned} FK_1 &= 4.3870; FK_2 = 4.6999; FK_3 = 0.6449; FK_4 = 0.5400; \\ FK_5 &= 0.0240. \end{aligned}$$

These values were used as the starting points for the optimization method to investigate if further reduction is possible.

6.3.2 Conventional direct search optimization method

The problem was next set up as an optimization problem. The objective function is defined as the sum of squares of weighted errors, i.e.

$$\text{Min } F(\underline{FK}) = \text{SSQWR}$$

subject to

$$a_i \leq FK_i \leq b_i \quad i = 1, \dots, NFK$$

The upper and lower limits on the force constants are as follows:-

$$4.0 \leq FK_1 \leq 6.5; 4.0 \leq FK_2 \leq 6.5 \quad 0.4 \leq FK_3 \leq 0.75;$$

$$0.4 \leq FK_4 \leq 0.8 \quad \text{and} \quad 0.02 \leq FK_5 \leq 0.085.$$

The direct search optimization method chosen was the simplex method and the algorithm has been described in the previous chapter. The main reason for choosing this method is that it does not require the evaluation of the derivatives of the function, which in this case

would be computationally expensive.

In the process of simplex expansion and contraction the values of the variables (force constants) may become negative or zero leading to fewer calculated frequencies. This prevents the calculation of the sum of squares of weighted error. For such cases the standard program has been modified in so far as a large value is assigned to the objective function whenever the above situation occurs. This large value indicates that no further steps should be taken in that direction. The same thing may happen if the force constants are widely out of their specified ranges, in which case extra frequencies will be calculated. A similar modification is applied to prevent this from happening.

Figures 6.1(a) and 6.1(b) show two such cases occurring in the simplex optimization process. In the first case a step is taken in the negative direction for FK_5 , and this produces seventeen frequencies. In the second case, FK_4 takes a large positive value which gives nineteen frequencies. In both cases further advances have been stopped in either direction by assigning a value of 10^8 to the SSQWR.

The first reduction in the size of the objective function occurs after twenty-six function evaluations. The SSQWR is reduced from 37038.83 to 4052.204. After sixty-six function evaluations the function is further reduced to 1430.65 with the force constants still within the acceptable ranges. The next reduction occurs after eighty one function evaluations with $SSQWR = 219.879$. This value was found to be acceptable by the chemist since the sum of squares of weighted error is not expected to be reduced to zero. The reason for this is because it uses a force field of a simple form. However, the mini-

mization process was carried out further and after ninety-six function evaluations the SSQWR = 199.291. After this stage further reduction did not appear to be significant and whenever a reduction in the size of SSQWR did occur the force constants were not within the acceptable range. Hence the process was terminated. In fact a substantial reduction had been made in the size of the sum of squares of weighted errors i.e. from a value of 37038.83 to a value of 199.291.

Figure 6.1(c) shows the final print out of the force constants, calculated frequencies and the sum of squares of weighted errors.

6.3.3 Plane analysis method

The last force constant FK_{NFK} , is known as the torsional force constant. The difference between the upper and lower bounds of this force constant is very small and it was found that a slight change in its value has a significant effect on the calculated frequencies and the sum of squares of weighted errors. In general, due to the nature of the problem which principally considers the vibrational entropy, changes in the low range force constants are more significant than those of high range force constants.

The first step of this method is to define the planar and non-planar variables. Taking into consideration the fact that the low valued variables, force constants, are the important ones, a choice of planar variables had to be made between FK_3 , FK_4 and FK_5 . In this case FK_3 and FK_4 were chosen as planar variables, and FK_1 , FK_2 and FK_5 were set at fixed values within the acceptable ranges.

Had FK_5 been chosen as a planar variable, a very small step size would be needed to draw the contours because the contours can only be considered for a small range of this variable. Hence, it is advisable to set this force constant at a fixed value initially, and

allow the user to choose a second value for it for the next slice.

To plot the contour of the function for the first plane slice, the non-planar variables were set at fixed values of $FK_1 = 4.5$; $FK_2 = 5.6$; and $FK_5 = 0.07$. The planar variables were given a range of $0.4 \leq FK_3$, $FK_4 \leq 1.0$. Figure 6.2(a) shows the contours of the first slice, and the minimum value of the function, which had been located visually, was $F = 778.528$.

To draw the next slice the ranges of planar variables were kept constant, but the value of non-planar variables were increased by one unit, i.e. $FK_1 = 5.5$; $FK_2 = 6.6$ and $FK_5 = 0.08$.

As before, the approximate minimum value of the function and the values of its corresponding variables were visually located. Since this problem is constrained and the user can adjust the values of non-planar variables, in particular the torsional force constant, it is considered adequate to perform just one linear search, and not the full search algorithm as defined earlier for unconstrained quadratic forms.

Figures 6.2(a) and 6.2(b) show the two contours from different plane slices. The minimum on the first contour has been located and has a value of $F = 778.528$. On the second contour the minimum has been found to have a value of 168.122 which is smaller than the value found by the simplex method, but because the value of FK_2 is greater than its upper bound a direct search was performed to find an acceptable point.

Figure 6.2(c) shows the result of the direct search with a sum of squares of weighted error of $SSQWR = 199.7551$ with force constants;

$$FK_1 = 5.2; FK_2 = 6.3; FK_3 = 0.54; FK_4 = 0.57; FK_5 = 0.077$$

Although this value is higher than the one located visually on the second contour the constraints are satisfied. However, if the user required a better accuracy to obtain the result a smaller step size could be taken to perform the search.

6.4 Comparison of the Methods

The hit and miss approach has the disadvantage that the change of value of one force field may result in bringing the value of one calculated frequency closer to the respective observed frequency, but at the same time changes the values of other frequencies, which may result in increasing the value of sum of square of weighted error. This approach was found to be very time consuming since small steps should be taken when changing the force constants and each force constant had to be modified in turn in order to obtain a reasonable result.

The basic simplex method does not take account of the problem constraints. The modification described recognises infeasibility and brings about a step back to feasibility. This leads to acceptable results but only after a lengthy process, since the variables are taken sequentially in numerical order, without regard to their contribution ratings.

The advantage of the interactive graphical plane analysis approach is that the user is in control of the running of the solution process. Since this problem is constrained the initial value of non-planar variables can be chosen by the user so that they will lie within the bounds. Once the contours are drawn the location of a few points within the contours by means of the cursor will give the user an in-

sight to what range of values could be taken for the variables.

The user immediately realizes when one of the variables does not satisfy the constraints and by modifying this variable an acceptable value of the function can be obtained.

Therefore it can be concluded that by using the plane analysis method the user can obtain a better understanding of the problem and participate fully in obtaining the result. A price is paid in terms of the number of function evaluations required. However, with experience, it is possible to get by with 'course' contours and linear searches selected initially, with higher resolution graphics brought in later if required.

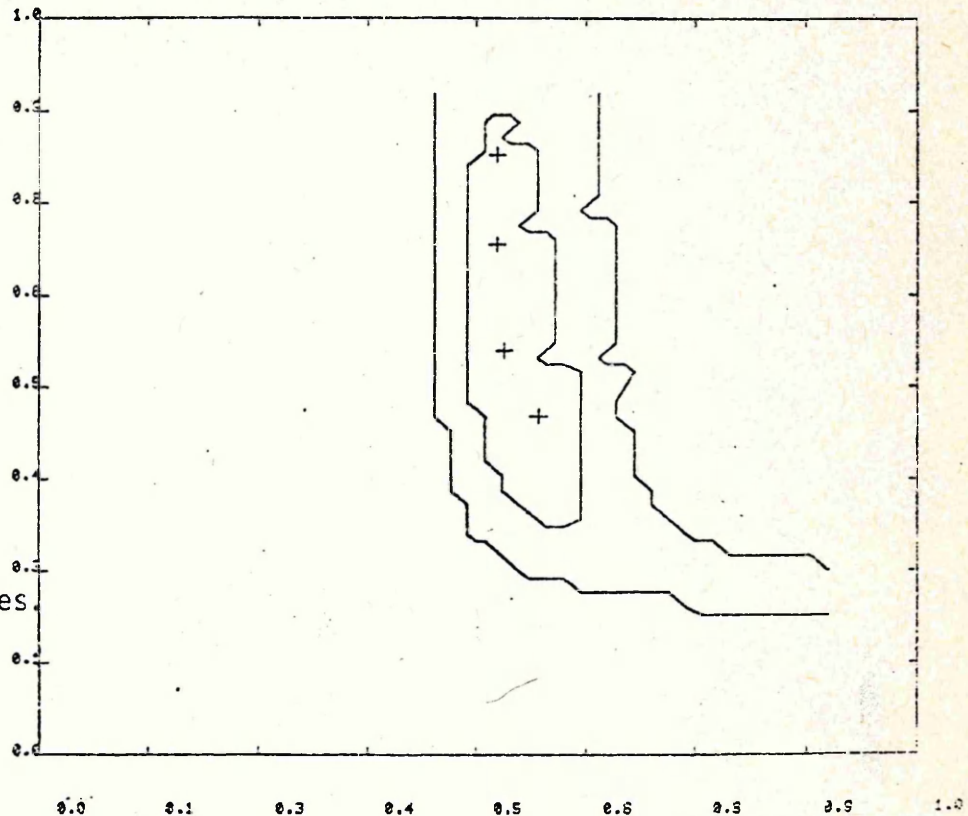
The approach should also be successful when applied to larger molecules than ethane, since once again the more significant force constants may be selected for variation and graphical display.

PRESS CP FOR CURSR
 PPFING E FOR NEW WINDO
 PPFING F FOR FINISH
 PPFING D FOR NEW PLANE

FIXED VARIABLES ARE
 FK1= 4.500 FK2= 5.000 FK5= 0.070

ERROR VALUE 849.877
 AXIAL VARLS
 FK3= 0.524 FK4= 0.815
 ERROR VALUE 800.237
 AXIAL VARLS
 FK3= 0.524 FK4= 0.694
 ERROR VALUE 778.528
 AXIAL VARLS
 FK3= 0.522 FK4= 0.550
 ERROR VALUE 201.156
 AXIAL VARLS
 FK3= 0.572 FK4= 0.461

Fig. 6.2a
 Contour on the
 Plane (x_3, x_4)
 x_1, x_2 and x_4 are
 set at fixed values

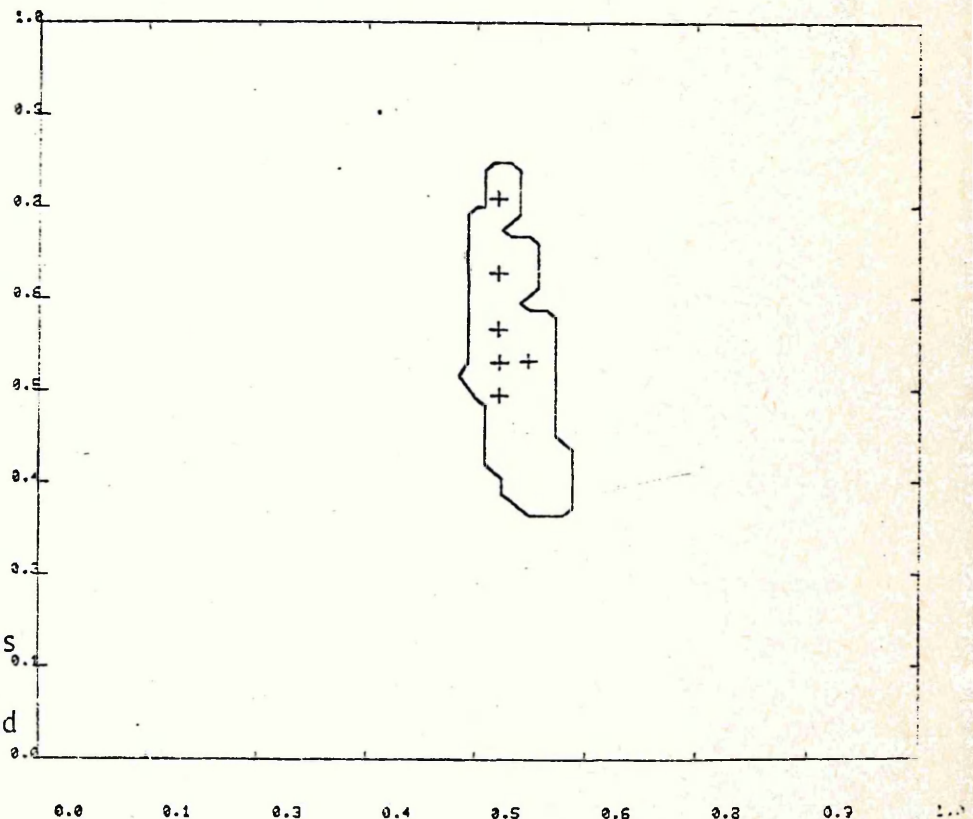


PRESS CP FOR CURSR
 PPFING E FOR NEW WINDO
 PPFING F FOR FINISH
 PPFING D FOR NEW PLANE

FIXED VARIABLES ARE
 FK1= 5.500 FK2= 5.000 FK5= 0.080

ERROR VALUE 235.263
 AXIAL VARLS
 FK3= 0.525 FK4= 0.762
 ERROR VALUE 191.108
 AXIAL VARLS
 FK3= 0.525 FK4= 0.661
 ERROR VALUE 176.140
 AXIAL VARLS
 FK3= 0.525 FK4= 0.524
 ERROR VALUE 180.508
 AXIAL VARLS
 FK3= 0.527 FK4= 0.495
 ERROR VALUE 168.122
 AXIAL VARLS
 FK3= 0.559 FK4= 0.541
 ERROR VALUE 172.885
 AXIAL VARLS
 FK3= 0.527 FK4= 0.539

Fig. 6.2b
 Contour on the
 Plane (x_3, x_4).
 The fixed values
 of x_1, x_2 and
 x_5 are increased
 by one unit.



PRESS G FOR ANEW WINDOW

PRESS C TO CONT. GRA NLU H
PRESS D TO DRAW NEW GRAPH & WINDOW

500.0

810.0

720.0

Fig. 6.2c
Direct Search Curve to
Calculate the Minimum
Value of Sum of Squared
of Weighted Error.

530.0

540.0

450.0

RESULT OF PROBLEM
MIN CAL OF VARS.
5.223 6.300
0.570 0.570
0.077

360.0

DIST= 0.70 F= 199.7551

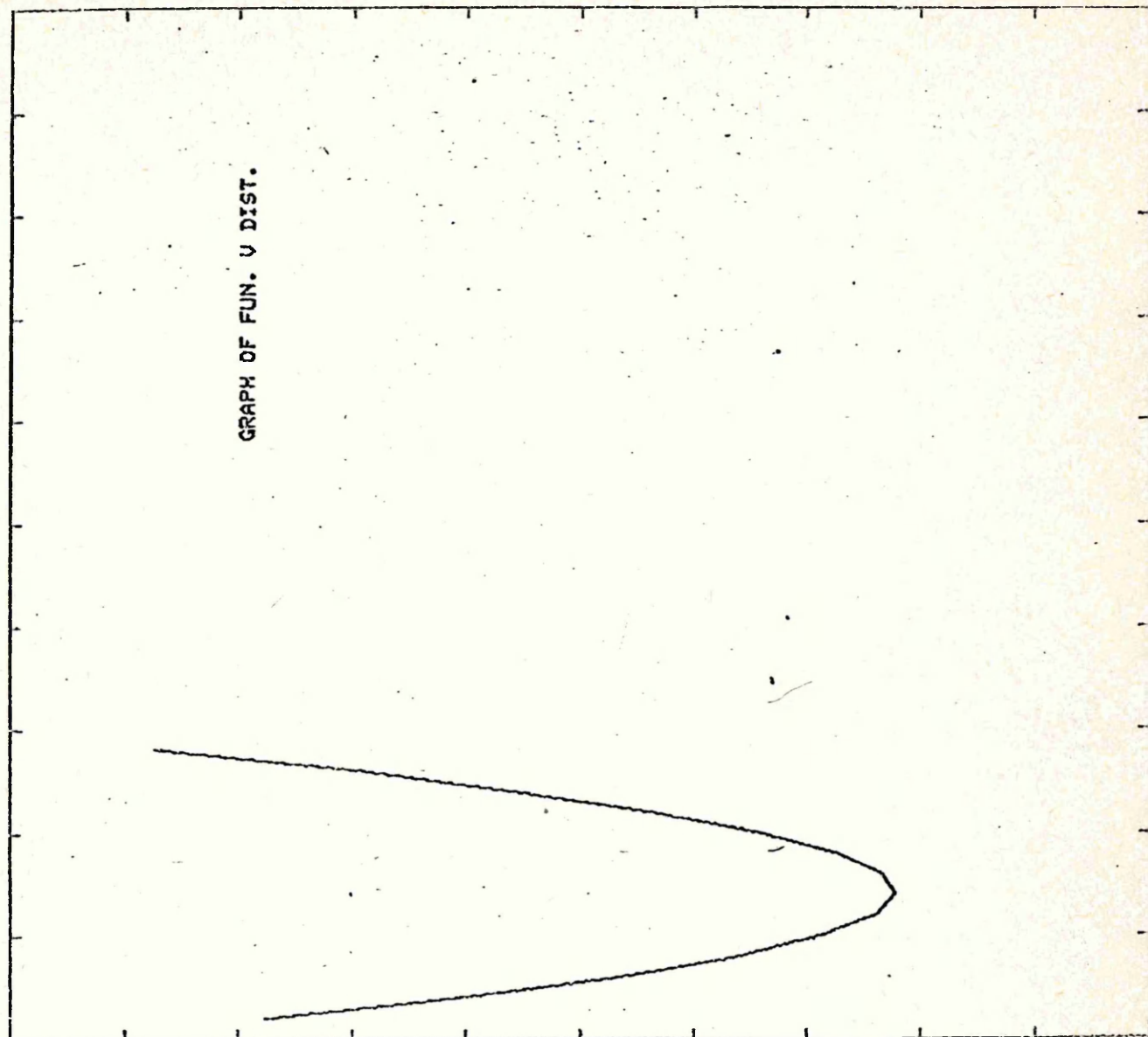
270.0

180.0

90.0

0.0

GRAPH OF FUN. V DIST.



0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

Appraisal and Conclusion

This thesis has brought together reviews of the current position in the fields of interactive computer graphics and non-linear optimization, and, through the development of and experiments with a comprehensive system, has shown that there are advantages to be gained by using interactive graphics to solve optimization problems.

The system developed on an IBM 370/135 linked to a Tektronix 4014 graphics display device has been used in several experiments.

The behaviour of existing routines on standard test problems in one and two dimensions has been illustrated graphically, and the system has been used as a means of interactively controlling conventional method for N-dimensional problems.

Experiments have also been conducted into the direct interactive visual optimization of low order N-dimensional functions and an algorithm has been derived for quadratic forms.

Finally, the system has been used to solve a constrained optimization problem arising in molecular vibrational analysis.

In each application described there has been clear evidence that interactive computer graphics provides an extra degree of insight into the behaviour of both existing and new techniques. Graphics is seen to enhance both analysis and presentation and often points the way ahead when other approaches falter. Consequently, it is believed that graphics has a substantial contribution to make in teaching, in method research and in practical problem solving.

With the equipment used here, the price paid for those advantages gained has generally been extended c.p.u. time, largely through the

use of an increased number of function evaluations. However, in terms of hardware, a faster central processor and a refreshed-type graphics display unit would undoubtedly enable more complex or larger problems to be tackled. On the software side, a faster contouring package would be of value, although the facility currently offered for user control of step size does enable economies to be made. Having said this, all of the problems described in this thesis have been capable of being dealt with in standard session times on the graphics system currently installed.

Turning now to further possible developments, there would seem to be ample scope for attempting solution of more problems involving constraints. Time has prevented the investigation of more than one such problem in this work, but the results were encouraging. Constrained problems are often more difficult to solve than unconstrained ones using conventional procedures and yet, with the aid of a capable graphics system, they could pose little extra difficulty using approaches along the lines of those proposed here.

Similarly, problems known to have several local minima would appear to offer interesting possibilities.

There is interest in graphics applications within optimization at other centres. A very recent paper by L G Birta (46) describes a package which limits its attention to the control of conventional non-linear optimization procedures, in a manner similar to that facility incorporated into the system and described earlier. The Numerical Optimization Centre at Hatfield Polytechnic is also active in this specific area and has investigated one-dimensional procedures using graphics.

The practical problem dealt with in this thesis was particularly suitable for the interactive graphical approach, and it seems likely that other such problems may exist within the fields of pure and applied science or operational research. It would be worthwhile to conduct investigation into the potential in other fields, for there are relatively few reported applications of interactive computer graphics to problem solving in general.

It is hoped that the work described in this thesis might inspire further, more ambitious applications of interactive computer graphics to the solution of non-linear optimization problems, whatever their source, and that the ideas expressed here might lead to the further use of interactive graphics as a means of deriving, testing and implementing methods for non-linear optimization.

APPENDIX

During the period of this work the author has attended the following conferences/meetings:

- (i) "New methods for non-linear constraints in optimization calculation" by Professor M J D Powell - Sheffield University, May 1977.
- (ii) Symposium on "Mathematical methods in computer graphics and design" - University of Leicester, September 1978.
- (iii) GINO Users Group meeting - University of Salford, July 1979.

The author has also regularly attended seminars held by the Institute of Mathematics and Its Applications and the Operational Research Society at Sheffield University and Sheffield City Polytechnic.

The guided reading list associated with this submission is as follows:

- (1) Fundamentals of optimization;
 - (a) "Methods of optimization" - G R Walsh (Wiley 1975).
 - (b) "Non-linear optimization techniques" - M J Box, D Davies, W H Swann, (Oliver & Boyd 1969).
 - (c) "Non-linear optimization" - L C W Dixon (EUP 1972).
- (2) State of art;
 - (d) "Towards global optimization", L C W Dixon, G P Szego (N Holland 1975).
 - (e) "A preliminary investigation of function optimization by a combination of methods" - D A Philips (Comp. J. 17, No. 1, 1974).
 - (f) "An algorithm for drawing the curve $f(X,Y) = 0$ " - D C Sutcliffe (Comp. J. 19, No. 3, 1976).

(3) Computer graphics;

- (g) "Principles of interactive computer graphics" - W N Newman and R F Sproull (McGraw-Hill 1973).
- (h) "Computer graphics - whence and hence" - J Potts (CAG Vol. 1, 1975).
- (i) "Applications of interactive computing in scientific environment" - T J Martin and A Sykes (On-line Conference 72).

The reading list also includes the journals of Computer Aided Design, Computer and Graphics, Computers and Education, and several others in Operational Research, Mathematics and Computing have been consulted.

REFERENCES

- (1) Jacobs, J F, "Practical Evaluation of Command and Control Systems", Report MTP-7, Bedford, Mass., MITRE Corporation (1965).
- (2) Sutherland, I E, "Man-Computer Graphics for Computer-Aided Design", Proc., IEEE, 54, No. 12 (1966).
- (3) Parslow, R D, "Computer Graphics - Applications and Case Histories", Plenum Press, (1970).
- (4) Green, R E, "Computer Graphics", Computer-Aided Design, Spring 1970.
- (5) English, W K, "Display-Selection Techniques for Text Manipulation", IEEE, (1967).
- (6) Wirth, N, "Euler - A Generalization of ALGOL", CACM, 6,9 (1963).
- (7) Ross, D T, "The AED Approach to Generalized Computer-Aided Design", Proc., ACM Conference (1967).
- (8) Parslow, R D and Green, R E, "Advanced Computer Graphics, Economics, Techniques and Applications", Plenum Press (1970).
- (9) Prince, M D, "Interactive Graphics for Computer-Aided Design", Addison-Wesley (1971).
- (10) Paterson, J W, "An Integrated c.a.d. System for an Architects Department", Computer-Aided Design 6, No. 1, (1974).
- (11) Ray-Jones, A, "Computer Development in West Sussex", Archt. J. 1, No. 2 (1968).
- (12) Beacon, G R, "Computer Modelling for Use in Architectural Design", Computer-Aided Design 10, No. 5 (1978).
- (13) Cardwell, D W "The Application of Computer Generated Graphs and Diagrams to the Development of Advance Nuclear Reactors", Oak Ridge National Laboratory (1970).

- (14) Mac Elroy, R D, "Computer Display and Manipulation of Biological Molecules", Computers and Graphics, Vol. 3,(1978).
- (15) Cooper, I, "Interactive Computer Graphics in Quantum Chemistry by Research", PhD Thesis, Keele University, (1974).
- (16) Hatvany, J, "CAD: The Next Ten Years", Computer-Aided Design 10, No. 6 (November 1978).
- (17) Newman, W N, and Sproul, R F, "Principles of Interactive Computer Graphics", McGraw-Hill (1973).
- (18) Zener, C, "A Mathematical Aid in Optimization Engineering Designs", Proc., Nat. Acad. Sci. 48 (1961).
- (19) Swann, W H, "Report on the Development of a New Direct Search Method of Optimization", Central Instrument Laboratory, ICI Research Note 64/3 (1964).
- (20) Powell, M J D, "An Efficient Method for Finding the Minimum of a Function of Several Variables Within Calculating Derivatives", The Computer Journal, 7 (1964) pp 155-162.
- (21) Kiefer, J, "Sequential Minimax for a Maximum", Proc. AMS 4, (1953).
- (22) Avriel, M, "Golden Block Optimization of Unimodel Functions Using an Unknown Number of Block", Chem. Eng. Dept. Rep., Stanford University, (1966).
- (23) Dixon, L C W, "Non-linear Optimization", EUP(1972).
- (24) Powell, M J D, "A Method of Minimizing the Sum of Squared Terms of Non-linear Function Without Calculating Derivation", The Computer Journal, 7 (1965).
- (25) Biggs, M C and Dixon, L C W, "The Advantages of Adjoint-Control Transformations when Determining Optimal Trajectories by Pontryagin's Maximum Principle", The Numerical Optimization Centre, The Hatfield Polytechnic, TR3 (1969).

- (26) Powell, M J D, AERE (Harwell) Computer Library Subroutine 'VA05A' (1969).
- (27) Forsythe, G E, and Motzin, T S, "Asymptotic Properties of the Optimum Gradient Method", Bull. Amer. Math. Soc. 57 (1951).
- (28) Fletcher, R, and Reeves, C M, "Function Minimization by Conjugate Gradients", The Computer Journal 7, (1964).
- (29) Hooke, R, and Jeeves, T A, "Direct Search Solution of Numerical and Statistical Problems", J. A.C.M. 8 (1961).
- (30) Rosenbrock, H H, "An Automatic Method of Finding the Greatest or Least Value of a Function", The Computer Journal 3, (1960).
- (31) Smith, C S, "The Automatic Computation of Maximum Likelihood Estimates", NCB/SD, Report No. SC 846/MR/40 (1962).
- (32) Powell, M J D, "An Efficient Method of Finding the Minimum of a Function of Several Variables Without Calculating Derivatives", The Computer Journal 7, (1964) pp 303-307.
- (33) Box, G E P, "On Experimental Attainment of Optimum Conditions", Journal of Royal Statistical Society B, 13, 1, (1951).
- (34) Spendly, W, and Hext, G R, "Sequential Application of Simplex Designs in Optimization and Evolutionary Operation", Technometrics 4, (1962).
- (35) Nelder, J A, and Mead, R, "A Simplex Method for Function Minimization", The Computer Journal, 5 (1965).
- (36) Ghani, S N, and Barnes, L, "Parameter Optimization for Unconstrained Objective Functions - a bibliography", Computer-Aided Design 4, No. 5, 1972.
- (37) Box, M J, "A Comparison of Several Current Optimization Methods and the Use of Transformation in Constraint Problems", The Computer Journal 9, (1966).

- (38) Dayhoff, M O, "A Contour Map for X-ray Crystallography", CACM, Vol. 6, (1963).
- (39) Robinson, E L, and Scarton, H A, "Contor - A Fortran Subroutine to plot smooth contours of a single valued 3-D surface", J. Computer Ph., 10, (1972).
- (40) Cattafava, G, and Le Moli, G, "Automatic Contour Map", CACM, Vol. 12, (1969).
- (41) Crane, C M, "Contour Plotting for Functions Specified at Nodal Points of an Irregular Mesh Based on an Arbitrary Two parameter co-ordinate System", The Computer Journal, 15, (1972).
- (42) McLaine, D M, "Drawing Contours from Arbitrary Data Points", The Computer Journal, Vol. 17, (1974).
- (43) Sutcliffe, D C, "An Algorithm for Drawing the Curve $f(X,Y)=0$ ", The Computer Journal, 19, No. 3, (1976).
- (44) Dixon, L C W, "The Safeguarded Cubic Technique", The Numerical Optimization Centre, The Hatfield Polytechnic T.R. 4 (1969).
- (45) Gwinn, W D, J. Chem. Phys., 55, (1971).
- (46) Birta, L G, and Guirgius, B, "An Interactive/Graphics Software System for Function Minimization", Computer and Graphics 4, (1979).